

CAPITOLO 2

LOGICA BINARIA

2.1 INTRODUZIONE ALLE VARIABILI BINARIE

Una variabile si dice binaria se può prendere soltanto due valori diversi o trovarsi in due stati diversi.

2.1.1 Frasi Logiche: Vero o Falso

Il giudizio di: **vero** o **falso** con cui una frase "logica" può essere caratterizzata ci fornisce un esempio di variabile binaria.

Es. 1) *La cassaforte di una banca si apre se così vogliono il direttore ed il vice direttore della banca.* Notiamo che questa frase risulta composta da due frasi più semplici che possono essere così espresse: 1a) *Il direttore della banca vuole aprire la cassaforte*; 1b) *Il vice direttore della banca vuole aprire la cassaforte*. Queste sono due frasi semplici che cioè *non* possono essere formulate mettendo assieme altre frasi o che è lo stesso *non* dipendono da altre condizioni e quindi costituiscono ognuna una quantità minima di informazione indipendente. Infatti se ci chiediamo se ogni una di essa è vera o falsa possiamo rispondere alla 1a) sapendo *solo* se il direttore vuole effettivamente aprire la cassaforte; 1b) sapendo *solo* se il vice direttore vuole effettivamente aprire la cassaforte. La veridicità della intera frase è subordinata alla **contemporanea veridicità** di ogni una di esse (**Prodotto Logico**); notare che se una delle due o entrambe le frasi sono false anche l'intera frase è falsa.

Per i nostri scopi è utile formulare quanto sopra, espresso in modo verbale, attraverso una corrispondenza, di ognuna delle frasi presenti nel discorso, con un simbolo diverso (variabile binaria) e cercare di esprimere attraverso di esse, e quindi in modo simbolico, il ragionamento fatto. Per il nostro caso possiamo utilizzare le variabili: A , B , C con la seguente corrispondenza:

- alla frase 1a) facciamo corrispondere la variabile binaria A ,
- alla frase 1b) facciamo corrispondere la variabile binaria B ,
- alla intera frase 1) facciamo corrispondere la variabile binaria C ,

Ogni una di queste variabili può assumere il valore V = vero o il *valore complementare* F =falso. La variabile C sarà $= V$ o $= F$ a seconda del valore che assumono le variabili A e B e a seconda di come l'intera frase dipende dalle frasi componenti. In altri termini possiamo dire che $C = f(A,B)$ cioè C è funzione delle variabili A e B . Il legame funzionale dipende alle condizioni imposte dal ragionamento. Allo scopo di ricavare questo legame funzionale utilizziamo una **tabella** detta **della verità**. Questa, poiché il numero delle variabili è piccolo e

poiché ogni una di esse può assumere solo due valori risulta sufficientemente sintetica e molto espressiva.

In una tabella della verità le colonne vengono intestate con i nomi scelti per ogni variabile binaria (A,B,C) ed i simboli V ed F indicano rispettivamente se la frase corrispondente è Vera o Falsa. Un doppio bordo verticale separa le variabili relative alle frasi semplici o *indipendenti* dalla variabile relativa alla frase composta *risultante o dipendente*. Nella tabella sono presentati tutte e quattro i casi possibili in cui le due variabili A e B si possono trovare e per ogni combinazioni delle variabili A e B viene indicato il valore che assume la variabile C (per la frase 1). Come è facilmente verificabile dal lettore la variabile

A	B	C
F	F	F
F	V	F
V	F	F
V	V	V

Tab 2.1 AND

$$C = V \text{ se } (A = V) \text{ e } (B = V)$$

il che corrisponde a quanto abbiamo già messo in evidenza.

Un altro modo per tradurre simbolicamente tale relazione è quello di introdurre un operatore che è, *definito dalla stessa tabella*, ed in questo caso si suole indicare con il simbolo algebrico di prodotto "x" o "." o nell'assenza dello stesso tra le due variabili logiche su cui opera. Esso prende il nome di operazioni di **prodotto logico**.

$$C = A \times B = A \cdot B = A B.$$

La tabella può anche essere letta come:

$$C = F \text{ se } (A = F) \text{ o } (B = F) \text{ o se } (A = F) \text{ e } (B = F).$$

Che corrisponde ad un altro modo di presentare la nostra frase senza alterarne il significato. In questo modo di ragionare invece di elencare i casi in cui una frase è vera (*ragionamento in logica positiva*) vengono elencati i casi in cui la frase è falsa (*ragionamento in logica negativa*). I due tipi di ragionamento sono detti **duali**. Notare che i due modi diversi di lettura della tabella possono essere ricavate una dall'altra con lo scambio contemporaneo di V con F, di e con o e con l'aggiunta dell'ulteriore caso che non può essere espresso utilizzando lo stato delle singole variabili.

Anche al tipo di ragionamento duale è possibile associare una relazione funzionale simbolica o equazione. Per distinguere però che ci stiamo riferendo ad un tipo di ragionamento duale del precedente dobbiamo usare, per ogni frase duale, una variabile che assume il valore falso se nella corrispondente frase diretta assumeva il valore vero e viceversa. Questo può essere messo in evidenza utilizzando la lettera usata precedentemente munita di un accento (variabile complementata).

Definiamo pertanto l'operatore "NOT" o di *complementazione* (per es. $A' = \text{NOT } A$) che opera sul valore della variabile nel modo rappresentato in Tab. 2.2. La frase duale è espressa nel seguente modo:

A	A'
V	F
F	V

Tab. 2.2 NOT

$$C' = A' + B'$$

in cui è stato introdotto l'operatore **somma logica** rappresentato con il simbolo algebrico "+", che traduce la parola o usata nella formulazione duale della frase (1). Ovviamente questo

operatore è rappresentato dalla stessa Tab. 2.1. L'apparente contraddizione che due operatori diversi possano essere rappresentati dalla stessa tabella viene rimossa se si nota che uno si riferisce alla logica diretta o **positiva** cioè *alla elencazione dei casi in cui una variabile è vera*, mentre l'altro si riferisce alla logica (duale) **negativa** cioè *alla elencazione dei casi in cui una variabile è falsa*.

Es. 2) Mario gioca a pallone se riceve l'autorizzazione da uno dei suoi genitori. Anche questa è una frase composta da frasi più semplici:

- 2a) Mario riceve l'autorizzazione dalla mamma (che indichiamo con la variabile A);
- 2b) Mario riceve l'autorizzazione da papà (che indichiamo con la variabile B).

A	B	C
F	F	F
F	V	V
V	F	V
V	V	V

Tab. 2.3 OR

La frase: Mario gioca a pallone (che indichiamo con la variabile C) è vera se A è vera o B è vera o A e B sono vere. Cioè

$$C = V \text{ se } (A = V) \text{ o } (B = V) \text{ o se } (A = V) \text{ e } (B = V).$$

Notare che questa volta la elencazione dei casi in cui la frase è vera ha la stessa espressione della frase (1) formulata in logica negativa. Pertanto l'operazione che questa frase definisce si chiamerà **somma logica** (per coerenza a quanto detto precedentemente) e sarà rappresentata simbolicamente con il segno algebrico " + ".

L'espressione simbolica è $C = A + B$

Indipendentemente dall'ultima osservazione, questa frase può essere espressa in forma tabulare come indicato dalla tabella della verità Tab. 2.3, che definisce l'operazione **somma logica o OR** tra due variabili in logica positiva.

La stessa frase, espressa in logica negativa, suona nel seguente modo: È Falso che Mario va a giocare a pallone se è *contemporaneamente falso* che il padre e la madre lo hanno autorizzato.

In altri termini $C = F \text{ se } (A = F) \text{ e } (B = F)$

La frase è ricavata dalla precedente con lo scambio di **V** con **F** e **o** con **e** ed è identica alla frase che ci ha permesso di definire l'operazione di prodotto logico in logica positiva. Notare inoltre che questa frase è suggerita dalla stessa Tab. 2.3 utilizzando i casi in cui la variabile C assume il valore F cioè usando la logica negativa.

Pertanto, coerentemente a quanto fatto in precedenza, utilizzando le variabile complementate, scriviamo che: $C' = A' \times B'$

Dall'analisi degli esempi 1) e 2) possiamo dire che una frase logica esprimibile mediante gli operatori di somma o di prodotto logico può essere rappresentata simbolicamente in due modi diversi essendo una l'espressione duale dell'altra. Queste frasi possono essere ricavate una dall'altra con la seguente regola: complementare le singole variabili e scambiare l'operatore \times con l'operatore +. In altri termini:

$$C = A \times B \Rightarrow C' = A' + B' \text{ o}$$

$$C = A + B \Rightarrow C' = A' \times B'$$

Due qualsiasi operatori che godono di questa proprietà sono dette uno duale dell'altro.

Es. 3): *La cassaforte della banca si apre se l'orario è tra le 8,00 e le 14 e se il direttore ed il vicedirettore sono contemporaneamente d'accordo ad aprirla.*

Questa frase è ricavata da quella dell'Es. 1 con l'aggiunta della condizione che l'ora deve essere tra le 8 e le 14. Questa modifica introduce essenzialmente una ulteriore condizione per cui per decidere se la frase è vera non basta chiederci se il direttore e vice direttore sono d'accordo ad aprirla ma dobbiamo anche chiederci se l'ora in cui la frase è pronunciata è o meno tra le ore 8 e le 14. In sostanza ora la nostra frase è composta da tre frasi semplici:

- 3a) l'ora è tra le 8 e le 14;
- 3b) Il direttore vuole aprire la cassaforte;
- 3c) Il vice direttore vuole aprire la cassaforte.

Se volessimo fare una rappresentazione simbolica di questa frase dovremmo definire 4 variabili binarie A, B, C, D con:

- A associata alla frase 3a),
- B associata alla frase 3b),
- C associata alla frase 3c)
- ed infine D associata all'intera frase 3.

A	B	C	D
F	F	F	F
F	F	V	F
F	V	F	F
F	V	V	F
V	F	F	F
V	F	V	F
V	V	F	F
V	V	V	V

Tab. 2.4 Es. 3

Una rappresentazione tabulare può essere vista in Tab. 2.4. Poiché tutte le possibili combinazioni dei valori delle variabili sono 8 otteniamo una tabella più lunga e complicata delle precedenti. Tuttavia l'espressione simbolica della frase può essere scritta come:

$$D = A \times B \times C \text{ oppure } D' = A' + B' + C'.$$

Il lettore verifichi che la interpretazione delle espressioni ora date come estensione di quelle precedentemente illustrate è conforme al problema enunciato.

Notare che l'Es. 3 differisce per l'introduzione di una terza variabile rispetto all'enunciato dell'Es. 1 e che ciò è riflesso nella costruzione della Tab. 2.4; le sue ultime 4 righe sono identiche a quelle della tabella dell'Es. 1, se noi trascuriamo di considerare la variabile A, ossia se non ci curiamo della variabile A o, per dirla in termini correntemente usati, se consideriamo A come una variabile **don't-care**. Una variabile è di tipo don't-care se essa non entra in gioco nel determinare la veridicità o meno dell'intera frase. Ciò si verificava nell'Es. 1 (non avevamo posto alcuna condizione per l'apertura della cassaforte). Se la richiesta della veridicità o meno della nostra frase viene fatta soltanto tra le ore 8 e le 14 (le ultime 4 combinazioni) o soltanto fuori questo orario (le prime 4 combinazioni) lo stato di A non entra mai in gioco e potremmo considerare o solo una metà o sola l'altra della tabella.

Si noti ancora che essendo la variabile A legata allo scorrere delle ore essa è vera o falsa in funzione del trascorrere del tempo.

Il lettore verifichi che un certo numero di frasi logiche, legate a situazioni completamente diverse, quando tradotte in forma simbolica diventano identiche a quelle già viste. Ciò vuol dire che l'uso di una espressione simbolica è molto più generale della frase da cui deriva. Siamo nella stessa situazione in cui l'uso di espressioni algebriche, nell'algebra usuale, ci permette di trattare nello stesso modo situazioni reali del tutto diverse. Per es. con l'espressione

$$c = a \times b$$

possiamo rappresentare sia il caso che si voglia conoscere il valore della variabile c (il costo di 2.5 Kg di patate) se $a = 2.5$ Kg e $b =$ il costo di 1 Kg di patate, sia la superficie di un rettangolo se a e b rappresentano le lunghezze dei suoi lati.

Es. 4) *Mario gioca a pallone se riceve l'autorizzazione da soltanto uno dei suoi genitori.* Questa frase è stata ricavata dall'Es. 2 con l'aggiunta della parola *soltanto*. Questa è una frase ancora composta da frasi più semplici: 4a) Mario riceve l'autorizzazione dalla mamma (che indichiamo con la variabile A); 4b) Mario riceve l'autorizzazione da papà (che indichiamo con la variabile B). Questa volta la frase: Mario gioca a pallone, che indichiamo con la variabile C , è vera se A è vera o B è vera e viene escluso il caso che A e B sono entrambe vere per la presenza della condizione *solo*.

La tabella della verità che traduce quanto ora detto è la Tab. 2.5. L'operazione che essa definisce viene chiamata **Somma Logica Esclusiva** e l'espressione simbolica che gli corrisponde è

A	B	C
F	F	F
F	V	V
V	F	V
V	V	F

Tab. 2.5 XOR

$$C = A \oplus B$$

Che si legge $C = V$ se $[(A = V) \text{ e } (B = F)]$ o $[(A = F) \text{ e } (B = V)]$

Usando gli operatori somma e prodotto logico può essere scritta:

$$C = (A \times B') + (A' \times B)$$

Se volessimo ricavare l'espressione duale di quella a logica positiva applicando quanto detto finora all'ultima espressione troveremmo che:

$$C' = (A \times B')' \times (A' \times B)' = (A' + B) \times (A + B')$$

Ragionando in logica negativa potremmo tuttavia dire: È Falso che Mario va a giocare a pallone se è *contemporaneamente falso o vero* che il padre e la madre lo hanno autorizzato ossia:

$$C = F \text{ se } [(A = V) \text{ e } (B = V)] \text{ o } [(A = F) \text{ e } (B = F)]$$

ed usando gli operatori somma e prodotto logico, può essere scritta:

$$C' = (A \times B) + (A' \times B')$$

Questa dovrebbe essere equivalente alla espressione trovata precedentemente. In effetti lo è ma noi non siamo ancora in grado di dimostrarlo poiché non sappiamo eseguire il prodotto di somme logiche.

L'operatore descritto verbalmente nell'ultima frase in corsivo e successivamente esplicitata in termini di somma di prodotti, viene chiamata **Equivalence** e si esprime con

$$C' = A \odot B = (A \oplus B)'$$

Che gode della proprietà di essere l'operatore **complementare** della Somma Logica Esclusiva e viceversa la Somma Logica Esclusiva è l'operatore **complementare** della **Equivalence** in quanto *per le stesse combinazioni di valori delle variabili indipendenti forniscono uno il valore complementare dell'altro per la variabile dipendente.*

Inoltre se a partire dalla definizione dell' **Equivalence**, si costruisce la sua tabella di verità usando la logica negativa, si vede subito che esso è anche l'operatore **duale** della Somma Logica Esclusiva cioè

$$C' = A \odot B = A' \oplus B'$$

Gli esempi finora illustrati hanno mostrato:

- le frasi logiche possono essere sempre scritte come espressioni coinvolgenti diversi operatori (con l'uso di varianti alle frasi già fornite o enunciando altre frasi è possibile introdurre degli altri che posseggono proprietà uguali o diverse da quelle già viste),
- che non tutti gli operatori sono indipendenti tra di loro e non godono delle stesse proprietà (per es. l'operatore + è duale all'operatore \times e viceversa ma gli operatori \oplus e \odot sono anche uno complementare dell'altro),
- le espressioni logiche simboliche assomigliano molto a delle vere e proprie relazioni funzionali, a prescindere da ciò che specificamente esse rappresentano.

Con queste premesse è facile convincersi che è auspicabile la ricerca della possibilità di definire una teoria generale ed astratta che, unificando le diverse relazioni, fornisca delle regole per un uso efficace sia delle varie espressioni che delle loro interrelazioni. Una teoria generale dovrà pure rendere semplice la trattazione di frasi logiche complesse (dipendenti da molte variabili). In effetti una tale teoria generale è stata ricercata e trovata nella seconda metà del secolo scorso (1854) da George Boole che sviluppò l'algebra che porta il suo nome (*algebra Booleana*) e di cui ci occuperemo più oltre.

2.1.2 Assegnazione di un valore numerico a una variabile binaria

Introduciamo ora una corrispondenza tra i due valori logici Vero e Falso e valori numerici tratti dall'algebra ordinaria. Poiché dobbiamo distinguere solo due situazioni diverse, possiamo scegliere di utilizzare i primi due numeri interi e cioè 0 ed 1. La corrispondenza può essere fatta attraverso una delle seguenti due scelte:

$$\text{Vero} \Leftrightarrow 1 \qquad \text{Falso} \Leftrightarrow 0 \qquad (1.1)$$

$$\text{Vero} \Leftrightarrow 0 \qquad \text{Falso} \Leftrightarrow 1 \qquad (1.2)$$

Con l'uso della (1.1) e della (1.2) troviamo che la tabella del Prodotto Logico o **AND** diventa

A	B	C
F	F	F
F	V	F
V	F	F
V	V	V

AND

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

assegnaz. (1.1)

A	B	C
1	1	1
1	0	1
0	1	1
0	0	0

assegnaz. (1.2)

Tab. 2.6 Tabella di verità AND con le due assegnazioni

la tabella della Somma Logica o **OR** diventa

A	B	C
F	F	F
F	V	V
V	F	V
V	V	V

OR

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

assegnaz. (1.1)

A	B	C
1	1	1
1	0	0
0	1	0
0	0	0

assegnaz. (1.2)

Tab. 2.7 Tabella di verità OR con le due assegnazioni

Se si confrontano le tabelle della verità (scritte con i simboli numerici 0 ed 1) dell'AND con l'assegnazione (1.1) e dell'OR con l'assegnazione (1.2), si vede che alla stessa combinazione di 0 ed 1 delle variabili indipendenti o di ingresso (A e B) corrisponde lo stesso risultato per il valore della variabile dipendente o di uscita. Lo stesso accade per il confronto tra l'OR e AND per le due diverse assegnazioni. Ciò non dovrebbe stupirci, visto che noi sappiamo che i due operatori sono uno duale dell'altro, e che, evidentemente, la diversa assegnazione equivale al cambiamento dalla logica positiva a quella negativa e quindi l'assegnazione:

$$\text{Vero} \Leftrightarrow 1 \qquad \text{Falso} \Leftrightarrow 0 \text{ determina la logica } \textit{positiva}$$

$$\text{Vero} \Leftrightarrow 0 \qquad \text{Falso} \Leftrightarrow 1 \text{ determina la logica } \textit{negativa}$$

Malgrado che entrambe le corrispondenze siano liberamente usate, noi, per fissare le idee, utilizzeremo la corrispondenza che porta alla definizione della logica positiva cioè vero \Leftrightarrow 1 e falso \Leftrightarrow 0.

Quanto ora messo in evidenza costituisce un esempio del così detto **principio di dualità** che al momento, utilizzando i valori 0 ed 1, enunciamo nel seguente modo:

Due operatori si dicono **duali** se scambiando, nella tabella della verità che definisce un operatore, 1 con 0 e viceversa, 0 con 1, si ricava la tabella della verità dell'altro operatore.

Due frasi o espressioni si dicono **duali** se una può essere ricavata dall'altra con lo scambio, nella tabella della verità che le rappresenta, di 1 con 0 e viceversa, 0 con 1.

Una variabile si dice **complementare** di un'altra se la prima ha il valore 1 quando l'altra assume il valore 0 e viceversa. L'operatore **NOT** produce il valore complementare della variabile di ingresso.

Un Operatore si dice **complementare** di un altro se, per la stessa combinazione delle variabili di ingresso, il risultato dell'uno è il complementare del risultato dell'altro.

Abbiamo già incontrato due operatori complementari e cioè la somma Logica Esclusiva o **Exclusive OR (XOR)** e l'**Equivalence (EQ)**.

Notare che se applichiamo l'operatore NOT al risultato di un qualsiasi operatore otteniamo il risultato complementato e quindi un operatore complementare. Pertanto per es.

$$\text{NOT}[\text{AND}(A,B)] = \text{NOT}(A \times B) = (A \times B)' = \text{NAND}(A,B)$$

$$\text{NOT}[\text{OR}(A,B)] = \text{NOT}(A+B) = (A+B)' = \text{NOR}(A,B)$$

$$\text{NOT}[\text{XOR}(A,B)] = \text{NOT}(A \oplus B) = (A \oplus B)' = A \odot B = \text{EQ}(A,B)$$

Ovviamente l'applicazione consecutiva di due NOT equivale a non applicarne alcuno

2.1.3 Rappresentazione grafica delle variabili binarie

Le variabili binarie possono essere rappresentate graficamente in modo molto espressivo attraverso delle mappe bidimensionali. In queste, ad ogni valore della variabile binaria, poiché può assumere solo due valori diversi, viene riservata metà della superficie della mappa. Per motivi pratici supporremo che il nostro spazio sia racchiuso da un rettangolo. Se dobbiamo rappresentare una sola variabile allora diviso il rettangolo in due parti uguali assegniamo una qualsiasi delle due metà al valore della variabile Vero o 1 e l'altra metà al suo valore complementare Falso o 0.

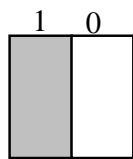


Fig. 2.1

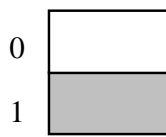


Fig. 2.2

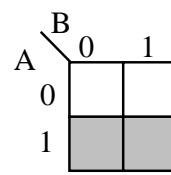


Fig. 2.3

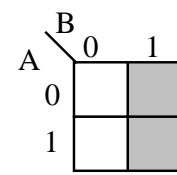


Fig. 2.4

Le Figg. 2.1 e 2.2. Mappa ad una Variabile

Mappa a due variabili

rappresentano la mappa di una sola variabile in cui lo spazio è stato diviso rispettivamente in modo verticale e orizzontale, all'area ombreggiata è stato assegnato il valore 1 e all'altra il valore 0. Le Figg. 2.3 e 2.4. rappresentano la mappa di due variabili. Lo spazio è stato diviso in quattro parti uguali. Alle variabili sono stati assegnati i nomi A e B e ad ogni loro valore è stato assegnato un semi rettangolo distinto.

La Fig. 2.3 rappresenta ombreggiato il rettangolo assegnato al valore $A = 1$ mentre la Fig. 2.4 rappresenta ombreggiato il rettangolo assegnato al valore $B = 1$.

Poiché il numero di tutte le combinazioni possibili di n variabili che possono assumere 2 valori, sono 2^n (nel nostro caso questo numero è 4) il rettangolo resta suddiviso in quattro parti uguali. Ognuna di queste rappresenta una particolare combinazione di variabili.

	B	0	1
A	0	00	01
	1	10	11

Fig. 2.5

	B	0	1
A	0	A'B'	A'B
	1	AB'	AB

Fig. 2.6

Mappa assegnazione variabili

Questa si ricava dalla combinazione che si ottiene considerando i numeri posti a fianco, a rappresentare i valori delle variabili, come le coordinate di una mappa stradale (vedi Fig. 2.5), oppure come mostrato nella Fig. 2.6 attraverso una sequenza di nomi di variabili ricavata assegnando al nome il valore 1 e al nome accentato il valore 0. Per es. alla combinazione $A'B$, che corrisponde alla sequenza dei valori 01 (0 per la variabile A ed 1 per la variabile B), viene assegnato il quadratino in alto a destra.

Definiamo **intersezione dei valori di due variabili** lo spazio costituito dai punti che appartengono sia al valore di una che dell'altra variabile.

Il quadratino $A'B$ rappresenta la intersezione dello spazio assegnato a $A = 0$ con quello assegnato a $B = 1$.

Definiamo **unione dei valori di due variabili** lo spazio costituito dai punti che appartengono o al valore di una o al valore dell'altra o ad entrambi

Utilizzando tali definizioni siamo in grado di dare una rappresentazione grafica degli operatori che abbiamo finora introdotto: AND e NAND in Fig. 2.7, OR e NOR in Fig. 2.8, XOR e EQ in Fig. 2.9. I quadratini ombreggiati rappresentano l'area che spetta alla

	B	0	1
A	0	A'B'	A'B
	1	AB'	AB

Fig. 2.7

	B	0	1
A	0	A'B'	A'B
	1	AB'	AB

Fig. 2.8

	B	0	1
A	0	A'B'	A'B
	1	AB'	AB

Fig. 2.9

combinazione dei valori delle variabili indipendenti che rendono il valore di quella dipendente pari a 1 logico (in applicazione della rispettiva definizione dell'operatore). Chiaramente agli operatori complementari corrisponderà l'area non ombreggiata.

Si vede chiaramente che l'operatore AND, es. $A \times B$, corrisponde all'operazione di intersezione delle variabili $A = 1$ e $B = 1$; l'operatore OR, es. $A + B$, corrisponde all'unione di $A = 1$ e $B = 1$ e che l'operazione NOT, es. $(A+B)'$, indica l'area complementare a quella corrispondente al risultato dell'operatore a cui è applicato, in questo caso $(A+B)$ e cioè $(A+B)' = A'B'$.

Le Fig. 2.10 ed 2.11 rappresentano rispettivamente le mappe relative a 3 e 4 variabili.

Da notare che il modo in cui le mappe possono essere definite non è unico cioè è possibile avere mappe con suddivisioni diverse da quelle presentate. Abbiamo scelto di utilizzare questa perché presenta delle mappe già nella forma che ci sarà utile successivamente.

In particolare, tale scelta, ha il vantaggio di permettere una facile individuazione delle coppie di quadratini caratterizzati dalla seguente proprietà: essi sono individuate da una sequenza di 0 ed 1 che differisce soltanto per lo scambio di uno 0 con 1 in una posizione qualsiasi o che è lo stesso: sono individuate da due

		BC			
		00	01	11	10
A	0	A'B'C'	A'B'C	A'BC	A'BC'
	1	AB'C'	AB'C	ABC	ABC'

Fig. 2.10

Mappa a 3 Variabili

		CD			
		00	01	11	10
AB	00	A'B'C'D'	A'B'CD	A'BCD	A'BCD'
	01	A'BC'D'	A'BCD	ABCD	ABCD'
	11	ABC'D'	ABC'D	ABCD	ABCD'
	10	AB'C'D'	AB'CD	AB'CD	AB'CD'

Fig. 2.11

Mappa a 4 variabili

configurazioni di variabili che differiscono per la complementazione di una sola variabile. Per es. 010 e 011 differiscono per lo scambio di 0 con 1 nella sola ultima cifra o che è lo stesso A'BC' e A'BC differiscono soltanto per complementazione della sola variabile C.

Questi quadratini sono individuabili, nelle mappe da noi costruite, come quelli che hanno un lato in comune e pertanto vengono detti *adiacenti*. Si noti che:

- per il caso di 1 e 2 variabili la definizione ora data di adiacenza è sufficiente ad individuarle tutte
- per il caso di 3 variabili bisogna aggiungere che la ricerca deve essere fatta dopo aver pensato di adagiare la mappa (considerata disegnata su un rettangolo di carta), su di un cilindro (di circonferenza di base pari alla dimensione maggiore del rettangolo stesso) con asse verticale
- per il caso di quattro variabili la ricerca deve essere fatta pensando di adagiare la mappa sia su un cilindro con asse orizzontale che con asse verticale.

Mappe con un numero di variabili >5 , che permettono di definire una regola opportuna per trovare i quadratini adiacenti, sono possibili ma, per la loro maggiore complicazione, risultano difficili da maneggiare. In tali casi si ricorre a metodi non grafici per es. per mezzo di un programma a calcolatore.

2.2 Elementi di circuiti elettrici con cui è possibile realizzare funzioni logiche

Abbiamo più volte detto che elettricamente è facile realizzare ed individuare variabili binarie. In quel che segue ne mostreremo alcune modi.

Interruttori: Un interruttore è un elemento circuitale che essenzialmente ha due stati stabili: in uno **si fa** attraversare da corrente e nell'**altro no**.

La sua rappresentazione grafica è quella di Fig. 2.12.

Lo stato stabile rappresentato nella Fig. 2.12(a) è

quello di **non** conduzione (infatti interrompe il cammino da A a B e l'eventuale corrente non può passare) o OFF. Lo stato di Fig. 2.12(b) è di conduzione o

ON. Se associamo allo stato OFF il valore 0 ed allo stato ON il valore 1 abbiamo realizzato praticamente una variabile binaria il cui valore (o stato) è memorizzato nello stesso interruttore. In altri termini mettere manualmente l'interruttore nello stato OFF equivale scrivere in questa memoria uno 0 e commutare l'interruttore nello stato ON equivale a cancellare in questa memoria lo 0 e scrivere un 1.

Con due interruttori una pila ed una lampadina costruiamo il circuito di Fig. 2.13. Questo è un *circuito logico o digitale*. Ai due interruttori sono associate le variabili binarie A e B (variabili indipendenti), alla lampadina è associata la variabile L questa è = 1 se accesa (è attraversata da corrente) e = 0 se spenta. L è funzione di A e di B secondo l'equazione $L = A \times B$ infatti ha il valore 1 se e solo se entrambi gli interruttori sono nello stato ON cioè $A = 1$ e $B = 1$.

Con gli stessi elementi collegati in modo diverso costruiamo il circuito logico 2.14. Adoperando la stessa convenzione del caso precedente abbiamo che $L = A + B$ in quanto la lampadina è accesa se l'interruttore A è ON o B è ON o entrambi sono ON.

Relè: Un altro modo di realizzare variabili binarie e circuiti che realizzano funzione logiche è attraverso l'uso di relè. Un relè è essenzialmente un commutatore elettrico comandato dalla magnetizzazione di una elettrocalamita. Quando nell'elettrocalamita non circola corrente il commutatore connette il terminale comune

(a in Fig. 2.15) ad uno dei due altri terminali (b in Fig. 2.15). Quando invece circola corrente il terminale a è collegato elettricamente con l'altro terminale (c in Fig. 2.15)

Il passaggio di corrente nell'avvolgimento dell'elettrocalamita può essere associato con la variabile binaria A che è = 1 se circola corrente e = 0 se non circola corrente. Per i terminali del relè b e c adottiamo la precedente convenzione;

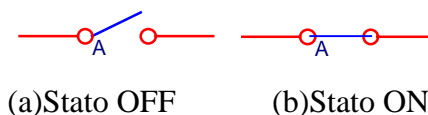


Fig. 2.12

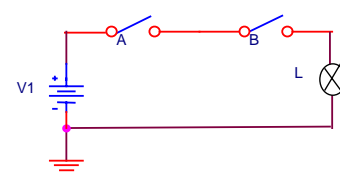


Fig. 2.13 AND $L = A \times B$

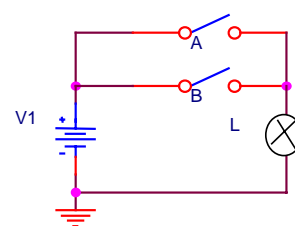


Fig. 2.14 OR $L = A + B$

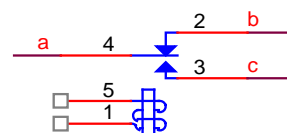


Fig. 2.15 Relè

- al terminale b (che è ON quando $A = 0$), associamo la variabile B .

Allora $B = 1$ quando $A = 0$ e $B = 0$ quando $A = 1$,

- al terminale c associamo la variabile C .

allora $C = 1$ quando $A = 1$ e $C = 0$ quando $A = 0$.

Notiamo immediatamente che $C = A = B'$ pertanto possiamo usare una sola variabile che indica contemporaneamente il passaggio di corrente

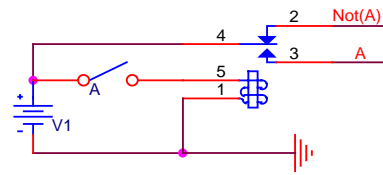


Fig. 2.16 Assegnazione Variabili

nell'avvolgimento e lo stato di conduzione dei due terminali del relè. Usiamo per es. la variabile A come è mostrato in Fig. 2.16. Questo elemento circuitale fornisce contemporaneamente il valore della variabile ed il suo valore complementare, ma a differenza del semplice interruttore non ha memoria del suo stato il quale dipende dal circuito che fornisce corrente all'avvolgimento. Poiché quando passa corrente nell'avvolgimento il relè commuta, il tipo di circuito elettronico che ne deriva è anche detto *circuiti di commutazione*.

Con due relè, una pila ed

una resistenza, costruiamo

il circuito di Fig. 2.17. Ai

due relè sono associate le

variabili binarie A e B

(variabili indipendenti),

alla resistenza è associata

la variabile L . $L=1$ se la resistenza è attraversata da corrente e $L=0$ in caso contrario. L è funzione di A e di B secondo l'equazione $L = A' \times B'$ infatti ha il valore 1 se e solo se entrambi i relè hanno gli avvolgimenti non attraversati da corrente.

Il lettore può facilmente trovare come modificare le connessioni perché la funzione binaria diventi $L = A \times B$; $L = A' \times B$ o $L = A \times B'$;

Adesso colleghiamo gli stessi componenti come rappresentato in Fig. 2.18. Usando per gli altri stati logici la

stessa convenzione del caso precedente abbiamo che: $L = A + B$ in quanto la resistenza è attraversata da corrente sia quando l'interruttore A è ON che quando B è ON o quando

lo sono entrambi. Il lettore provi a cambiare il collegamento della resistenza ai terminali A , A' , B e B' rispetto a quelli di

figura e provi a determinare la funzione logica realizzata per ogni combinazione.

I circuiti di Fig. 2.17 e 2.18 suggeriscono una generalizzazione di rappresentazione dei circuiti binari, logici o di commutazione che ci permetterà di non fare più riferimento al particolare tipo di realizzazione delle funzioni stesse.

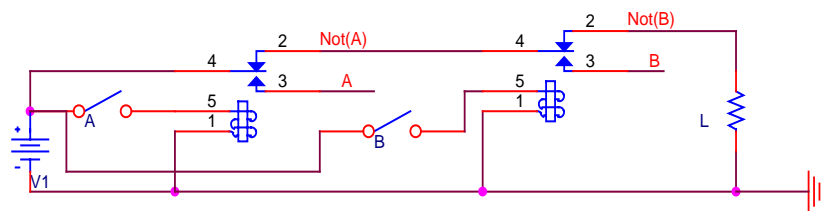


Fig 2.17 $L=A' \times B'$

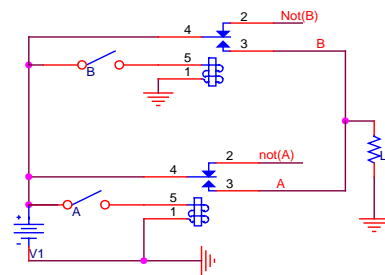
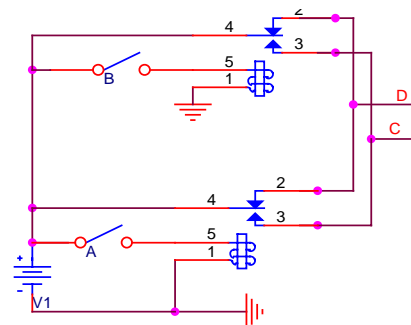


Fig. 2.18 $L=A+B$

La Fig. 2.18a rappresenta una modifica del circuito di Fig. 2.18 in cui le variabili binarie A e B sono state assegnate agli interruttori degli avvolgimenti dei relè; lo stato del relè è determinato dallo stato del suo interruttore. Poiché questo dipende dall'azione sull'interruttore esse sono delle variabili indipendenti o di ingresso. I terminali dei relè che normalmente fanno contatto (normali) e non (il cui stato dipende dallo stato dei relè A e B) sono stati connessi, a coppia, tra di loro. Essi sono le variabili dipendenti o di uscita. La variabile D è stata assegnata ai terminali normalmente in contatto e C agli altri. Ogni una di esse è una variabile binaria come quella determinata dallo stato degli interruttori in quanto che, se si collega l'avvolgimento di un nuovo relè a C o D, il relè sarà eccitato o meno a seconda se la variabile sarà = 1 o 0. Le funzioni binarie o logiche realizzate da questo circuito sono $C = A + B$ che abbiamo anche chiamata OR (logica positiva) e $D = A' + B' = (A \times B)'$ che abbiamo anche chiamata NAND.

Fig. 2.18.a $D=A'+B'$; $C=A+B$

Notare che nel determinare le connessioni dei contatti dei relè abbiamo limitato la duttilità del circuito: può essere utilizzato solo per le funzioni trovate. Questo è compensato dal fatto che l'utente se invece di comperare i singoli elementi circuitali compera l'intera funzione gran parte del lavoro (manuale) di collegamento è fatto e quindi si può concentrare sull'uso della funzione logica anziché sulla sua realizzazione. Un altro vantaggio deriva dal fatto che l'uscita elettrica (C o D nel nostro esempio) è in grado di pilotare, con il suo stato logico, altri elementi di circuito logico (nel nostro caso l'avvolgimento di un altro relè). Questo concetto è estremamente importante in quanto ci fornisce la possibilità di utilizzare l'uscita logica di un circuito binario come dato di ingresso di un altro circuito binario. In questo modo possono essere costruite funzioni logiche più complesse.

Questo tipo di approccio è risultato tanto positivo che, in effetti, sono disponibili sul mercato tutte le funzioni logiche normalmente richieste per la realizzazione di circuiti logici complessi: l'utente non deve fare altro che gli opportuni collegamenti.

Le funzioni logiche predefinite sono presentate come delle scatole chiuse ed inaccessibili se non ai terminali di collegamento e di alimentazione (chiaramente identificati). Ovviamente in esse non è presente la pila (per ovvi motivi) ma sono indicati i terminali a cui l'alimentazione esterna deve essere collegata.

Questa scelta ha portato ad una semplificazione della rappresentazione grafica di funzioni logiche in quanto non si è più interessati ai particolari realizzativi delle singole funzioni ma, al loro uso allo scopo di realizzare funzioni logiche più complesse. La funzione logica di Fig. 2.18a può essere benissimo rappresentata dal

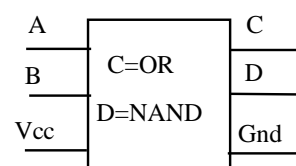


Fig. 2.18b

grafico di Fig. 2.8b, in cui sono esplicitate soltanto i piedini di ingresso, di uscita ed il nome delle funzioni (riferite alle variabili normali).

2.2.1 Uso dei Transistor CMOS come interruttori

Un altro elemento circuitale (di cui è facile capire il funzionamento logico) utilizzabile per la realizzazione di funzioni logiche sono i transistor CMOS. Questi hanno quattro terminali (*gate*, *source*, *drain* ed un quarto terminale che nel prosieguo ignoreremo). Il funzionamento logico di un transistor CMOS è quello dell'interruttore. Perché la corrente possa passare dal terminale source al drain l'interruttore deve essere chiuso. I terminali source e drain, fino a che ci interessiamo soltanto di segnali digitali possono essere pensati equivalenti (non ci interesseremo quindi a distinguerli con delle etichette)

Usando il terminale di gate faremo commutare il transistor on (chiuso o conduttore) o off (aperto o non conduttore). Ci sono due specie di transistor CMOS: *n-channel* e *p-channel*. Diremo che il livello logico ad un terminale è all'1 logico se la tensione è VDD (la tensione di alimentazione del transistor, +5Volt), diremo che allo 0 logico se la tensione è VSS (la tensione di riferimento, 0 Volt). Un 1 logico in gate rende un transistor *n-channel* conduttore, mentre per rendere conduttore un transistor *p-channel* bisogna fornire in gate uno 0 logico.

Nella rappresentazione grafica un transistor *p-channel* ha un cerchietto sul gate. In generale un

cerchietto, premesso ad un qualsiasi terminale, indica che il livello attivo è invertito rispetto a

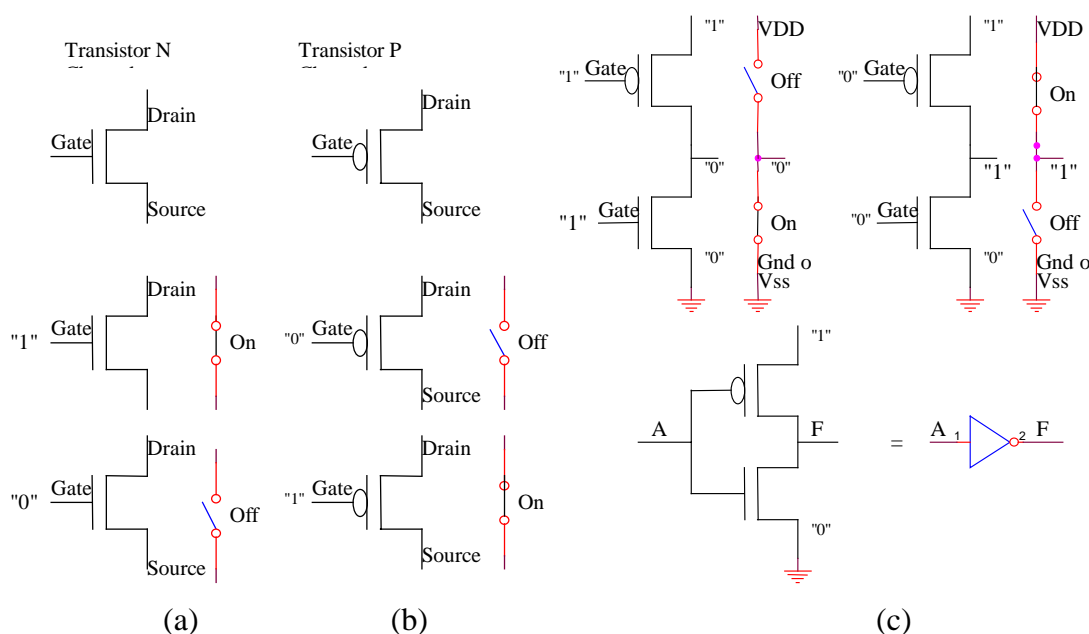


Fig. 2.19: Il transistor CMOS come interruttore: (a) Un transistor *n-channel*. (b) Un transistor *p-channel*. (c) Un *inverter* CMOS ed il suo simbolo (un triangolo equilatero ed un cerchio)

quello normale della logica positiva (1 logico) e quindi in questo caso ci fa capire che il livello necessario a che il p-channel diventi conduttore è lo 0 logico. La Fig. 2.19 ne fornisce una rappresentazione grafica.

Se connettiamo un transistor n-channel in serie con un p-channel come mostrato in Fig. 2.19 (c), formiamo un inverter.

La Fig. 2.20 mostra come è possibile con l'uso di quattro transistor formare una funzione NAND a due ingressi, Fig. 2.20(a) e una funzione NOR a due ingressi, Fig. 2.20(b)

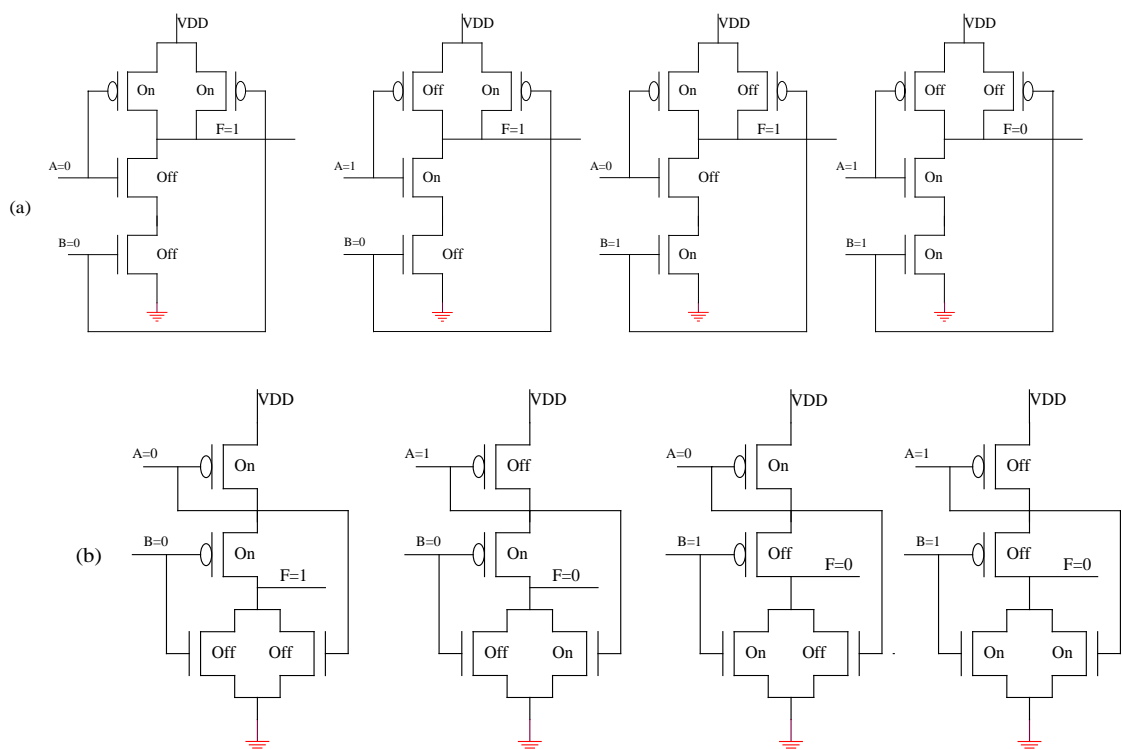


Fig. 2.20: Logica CMOS.

(a) Funzione NAND a due ingressi $F=(AB)'$. (b) Funzione NOR a due ingressi $F=(A+B)'$

2.3. Famiglie logiche: generalità

Bisogna precisare che, seppure l'utente di circuiti elettronici (realizzanti funzioni logiche), non è interessato ai dettagli realizzativi degli stessi, non può fare a meno di conoscere, per un loro uso appropriato, quanto di seguito elencato:

- quale è il valore di tensione di alimentazione del circuito
- quanto è la corrente e la tensione minima e massima che si deve applicare ai terminali di ingresso per realizzare lo stato logico 0 o lo stato logico 1

- quali devono essere le caratteristiche temporali delle transizione tra i livelli logici da 1 a 0 e da 0 ad 1 in ingresso
- quali sono i valori di tensione e corrente (minima e massima) che si può prelevare dai terminali di uscita nei diversi stati logici (fan-out)
- quali sono le caratteristiche temporali delle transizione tra i livelli logici da 1 a 0 e da 0 ad 1 in uscita (notiamo incidentalmente che essendo possibile utilizzare le uscite come variabili di ingresso di altri circuiti logici queste debbono avere caratteristiche essenzialmente equivalenti a quelle richieste per le variabili di ingresso sia nelle caratteristiche di commutazioni che di livelli logici).
- quale è il ritardo temporale tra l'istante in cui avviene una transizione in ingresso e l'istante in cui è disponibile la risposta in uscita (tempo di propagazione)
- quale è la frequenza massima con cui le variabili di ingresso possono cambiare senza che il circuito cessi di funzionare come dichiarato dal costruttore

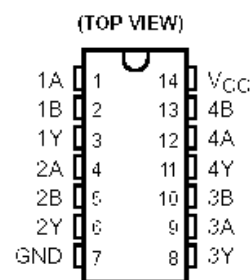
La risposta a queste domande dipende dal particolare tipo di realizzazione elettronica delle funzioni logiche o come si dice dalle famiglie logiche.

Nel corso dell'ultimo trentennio sono state realizzate varie famiglie logiche che vengono individuate con delle sigle che derivano normalmente dalla tecnologia usata. Diciamo subito che la famiglia logica realizzata con relè è stata la prima ad essere disponibile, ma subito abbandonata, per l'avvento della tecnologia dei semiconduttori, a causa del grande ingombro e richiesta di potenza (noi li abbiamo riportati come esempio di realizzazione, per il loro valore storico e perché sono facili da capire; richiedono concetti elementari certamente noti al lettore dai programmi liceali). Tra le famiglie logiche a semiconduttore ricordiamo la DTL (Diode Transistor Logic), la RTL (Resistor Transistor Logic), la TTL (Transistor Logic), la CMOS (Complementary Metal Oxide Semiconductor). In generale ogni famiglia logica ha delle sottofamiglie che li differenziano per es. per la richiesta di potenza (Low Power o High Power), per la velocità di risposta (Low Speed, High Speed) e per altro.

L'elencazione completa di tutte le famiglie logiche è al di là degli scopi del presente lavoro. Pertanto si rimanda il lettore alla consultazione dei relativi cataloghi. Ricordiamo che in questi, assieme alla elencazione delle caratteristiche elettriche generali della famiglia e delle sottofamiglie, sono elencati, per ogni funzione logica disponibile, le caratteristiche funzionali specifiche del circuito elettronico che le realizza e la "*Pin Function*"; cioè la corrispondenza tra la posizione di un piedino o terminale nel *chip* e la sua funzione logica come per es. il piedino a cui va collegato il terminale positivo della tensione di alimentazione o V_{cc} , il terminale negativo della tensione di alimentazione o Ground (GND), i piedini di ingresso e di uscita.

I circuiti logici sono impaccati in contenitori di dimensioni standard con un numero di piedini variabile: 8, 14, 16, 22, etc. Il contenitore o package viene chiamato genericamente CHIP e la sua forma standard più diffusa è quella detta "Dual in Line" rappresentata in Fig. 2.21. In esso

si individua una indicazione "Top View" che significa che la sequenza di numerazione dei piedini si riferisce al chip visto dall'alto. Un riferimento sta indicare la posizione di inizio della numerazione (Questa non si trova scritta sul contenitore ed è individuabile soltanto per mezzo delle indicazioni topologiche richiamate). Una sigla che ne indica la famiglia (in questo caso TTL) ed una indicazione numerica per es. 74LS00N che ne specifica la funzione e la sottofamiglia (questa corrisponde alla funzione NAND della sottofamiglia Low Power Schottky). Il volume occupato dal circuito elettronico che realizza la funzione logica, dovuto alla odierna possibilità di miniaturizzazione è estremamente più piccola del contenitore (le cui dimensioni sono piuttosto dettate dalla possibilità di accesso ai singoli pin). In un package c'è spazio per un certo numero di copie di circuiti realizzanti la stessa funzione logica ciascuno dei quali ricevono ingressi e forniscono uscite attraverso piedini dedicati e differenti. Nel chip indicato in Fig. 2.21 ci sono 4 NAND. Il numero di copie presenti dipende essenzialmente dal numero di piedini del chip (in questo caso sono 14 così destinati: piedino 14 a Vcc; 7 a Gnd; gli altri 12 piedini, a gruppi di 3, sono destinati ai due ingressi e all'uscita di ogni NAND; 1,2 ingressi e 3 uscita; 4,5 ingressi e 6 uscita; 9 e 10 ingressi e 8 uscita; 13 e 12 ingressi e 11 uscita.)



74LS00N

Fig. 2.21

2.3.1 Definizione dei livelli logici

Tra le informazioni più importanti che definiscono le diverse famiglie logiche ricordiamo i valori nominali di tensione e/o di corrente che definiscono i livelli logici 1 e 0. Per indicazioni precise si rimanda al catalogo che descrive la famiglia logica scelta, qui ci limitiamo alla presentazione generale delle caratteristiche dei TTL, CMOS e delle nuove famiglie che lavorano a bassa tensione. La nascita

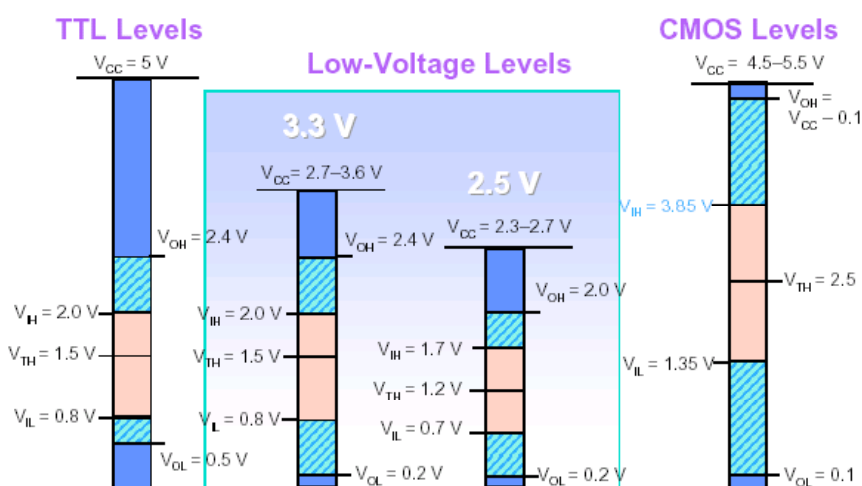


Fig. 2.22

di quest'ultime è stata determinata dall'aumento delle frequenze di lavoro. Infatti, per es. la famiglia CMOS, che è pilotata in tensione, dissipa la maggior parte della potenza alle

transizione dovuto al fatto che quando queste si verificano ci sono delle capacità parassite che debbono essere caricate o scaricate. Dalla Fig. 2.22 vediamo che per ogni famiglia sono definite due valori di tensione:

- 1) per l'ingresso (a sinistra della barra verticale) e
- 2) per l'uscita a destra della stessa barra verticale.

Noi descriveremo in dettaglio, basandoci sulla Fig. 2.22, solo la famiglia TTL poiché la lettura della figura per le altre famiglie è identica.

- La tensione di alimentazione è 5 Volt,
- Il livello di uscita, V_{OH} , che corrisponde in logica positiva a 1 logico, può variare da un massimo di 5 Volt ad un minimo 2,4V. Questo valore dipende soprattutto dal numero di ingressi equivalenti che l'uscita deve pilotare. Infatti il circuito equivalente Thevenin dello stadio di uscita può essere pensato come un generatore di tensione di 5V con una resistenza interna in serie. È su di essa che cade la differenza di potenziale che manca ai terminali di uscita quando gli ingressi vengono pilotati.
- Il livello di uscita, V_{OL} , che corrisponde in logica positiva a 0 logico, può variare da +0,5Volt ad un minimo 0V
- Il livello di ingresso, V_{IH} , che corrisponde in logica positiva ad 1 logico, può variare da un minimo di 2V ad un massimo di 5V. Il che significa che una tensione di ingresso entro questa regione viene interpretata dal circuito come un uno logico.
- Il livello di ingresso, V_{IL} , che corrisponde in logica positiva a 0 logico, può variare da minimo 0V a massimo +0,8V. Il che significa che una tensione di ingresso entro questa regione viene interpretata dal circuito come uno zero logico. Notare che questi livelli di ingresso sono opportunamente definiti (*interni* agli intervalli delle tensione in uscita per ciascun livello logico) in modo che, se vengono pilotati dalle uscite delle funzioni logiche della stessa famiglia, gli stati logici vengono interpretati correttamente.

Dall'analisi delle quattro barre di Fig. 2.22 si vede che funzioni logiche appartenenti alle diverse famiglie non possono essere immediatamente miscelate. Infatti tra di esse non esiste una relazione di compatibilità assoluta dei livelli di tensione in ingresso e in uscita. Per risolvere questi problemi sono disponibili, per ogni famiglia, dei circuiti "*Level Shifter*" il cui compito è soltanto quello di accettare correttamente i livelli logici della famiglia di appartenenza e di trasformarli in uscite in modo da essere compatibili con quelli della famiglia logica di destinazione.

Osservando la Fig. 2.22 si riconosce la presenza di un intervallo di tensione a cui non può essere attribuito né il valore zero né il valore uno: *questo un intervallo è proibito*. In effetti quando accade che una tensione in uscita fa una transizione a livelli bassi o a livelli alti, essa deve necessariamente attraversare la zona proibita. Se l'attraversasse lentamente, rispetto ai tempi di risposta del circuito che pilota, vi sarebbe il rischio che i vari livelli di tensione, che man mano assumerebbe, possano essere riconosciuti o come zero o come uno

indifferentemente. Questa indecisione sarebbe presente in uscita con delle oscillazioni di livello di tensione, rispecchianti l'elaborazione logica della interpretazione dei livelli nella zona proibita. Se invece il tempo che impiega ad attraversare la zona proibita è inferiore al tempo di reazione allora, ammesso che ci possa essere una interpretazione sbagliata del livello, la sua elaborazione non si propagherà in uscita prima che il livello di ingresso supera la zona proibita e quindi l'uscita pervenga allo stato di uscita definitivo. Questo spiega perché vengono anche definite per ogni famiglia, i tempi minimi delle transizioni tra livelli logici. È ovvio che le caratteristiche delle variazioni temporali in uscita debbono essere compatibili con quelle necessarie agli ingressi delle funzioni della stessa famiglia logica.

2.3.2 Simboli standard per i circuiti elettronici che realizzano funzioni logiche

I circuiti logici digitali sono detti anche semplicemente *circuiti logici* perché, attraverso appropriati ingressi, stabiliscono cammini di manipolazione logica. *Qualsiasi informazione, sia essa destinata al calcolo che al controllo, può essere realizzata per mezzo del passaggio di segnali binari attraverso varie combinazioni di circuiti logici, ciascun segnale rappresentando una variabile e trasportando un bit di informazione.* Circuiti logici che fanno operazioni logiche di AND, OR, e NOT sono mostrati con i loro simboli in Fig. 2.23. Questi simboli sono standard e più sofisticati di quello di Fig. 2.18b in quanto ad ogni funzione logica è dedicato un simbolo diverso. Ciò permette di non appesantire i grafici di circuiti logici con la indicazione della funzione dei singoli blocchi in quanto la funzione è di per se evidente. Questi circuiti sono chiamati anche porte logiche o *gates*. Qualche volta il circuito NOT è chiamato *circuito invertitore* poiché inverte un segnale binario.

In Fig. 2.23 sono presentati anche i simboli dei circuiti logici NAND, NOR, che logicamente consistono rispettivamente di un circuito AND o OR seguito da un NOT. Questo è rappresentato dal cerchietto posto a destra. Notare che l'equivalenza è soltanto logica e non è detto che circuiti NAND o NOR siano effettivamente realizzati come detto e quindi più complicati. Questo è evidente dai circuiti già visti, realizzati con relè e con CMOS,

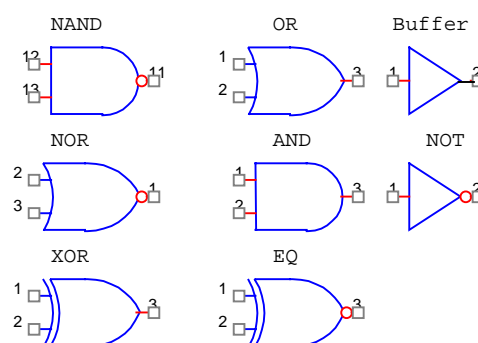


Fig. 2.23

inoltre normalmente, per i circuiti logici a semiconduttore, è vero il viceversa. Abbiamo voluto evidenziare questo fatto perché il lettore capisca che quando si parla di equivalenza, dal punto di vista logico tra due circuiti, non ci si riferisce mai alla loro particolare realizzazione ma alla loro funzione.

2.3.3 Fan_out

Notiamo che in Fig. 2.23 abbiamo un simbolo chiamato *Buffer*. Questo simbolo assomiglia al NOT ma non ha il cerchietto. Il motivo è che *Non Inverte* lo stato logico dell'ingresso ed è un circuito ausiliare al problema che sorge quando uno stesso segnale logico deve essere connesso a un certo numero di ingressi di porte logiche diverse.

Premesso che: *Il numero massimo di ingressi che l'uscita di una gate può pilotare prende il nome di fan_out*, notiamo che l'ingresso di una gate richiede, al circuito che lo pilota, una tensione, il cui valore è legato allo stato logico, ed una corrente adeguata e quindi della potenza. L'uscita di un qualsiasi circuito logico è in grado di fornire fino a una data potenza massima (che è indicata nel catalogo della famiglia come fan_out), superata la quale i livelli logici non sono più rispettati e perciò non funziona più nel modo dichiarato. In questi casi alcuni ingressi vengono pilotati direttamente dall'uscita della gate, tra cui quello di un *buffer*, e gli altri attraverso l'uscita del buffer stesso. Questo, se è qualificato "*buffer di potenza*", ha un fan_out maggiore di una qualunque gate della famiglia.

2.3.4 Rappresentazione grafica di segnali logici in funzione del tempo (TIMING)

Come già messo in evidenza la verità o falsità di una frase e quindi il segnale logico che lo rappresenta, può essere funzione del tempo. È opportuno per "vedere" l'evoluzione temporale fare delle rappresentazioni grafiche.

Questi grafici temporali o più brevemente "Timing", dall'inglese, sono realizzati rappresentando sull'asse orizzontale il tempo, in scala lineare, e sull'asse verticale il valore logico della variabile (0 o 1). Poiché il numero delle variabili coinvolte in un problema può essere più di una e tra di loro esiste certamente una correlazione temporale, i grafici relativi ad ogni una di esse sono

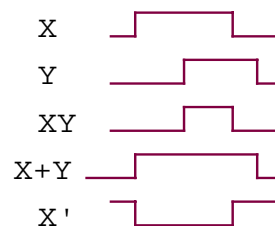


Fig. 2.24

disegnati sovrapposti: spostati verticalmente uno dall'altro e coincidenti orizzontalmente (in modo che l'origine dei tempi sia uguale per tutti). Il grafico di Fig. 2.24 esemplifica quanto ora detto. Il nome delle variabili relativo ad ogni segnale logico distingue i diversi grafici e/o mettono in luce l'eventuale relazione logica (OR, AND, etc.) attraverso l'equazione che le rappresenta. Notare che poiché spesso non interessano i valori temporali assoluti ma la correlazione tra i vari segnali, l'asse dei tempi non viene esplicitamente disegnato.

PROBLEMI

- P2.1 Scrivere almeno 5 frasi logiche composte da due frasi semplici che, quando espresse con variabili binarie diano luogo ad una tabella di verità di tipo AND per la enumerazione delle combinazioni per cui la frase è vera
- P2.2 Scrivere almeno 5 frasi logiche composte da due frasi semplici che, quando espresse con variabili binarie diano luogo ad una tabella di verità di tipo OR per la enumerazione delle combinazioni per cui la frase è vera
- P2.3 Scrivere almeno 5 frasi logiche composte da due frasi semplici che, quando espresse con variabili binarie diano luogo ad una tabella di verità di tipo XOR per la enumerazione delle combinazioni per cui la frase è vera
- P2.4 Scrivere almeno 5 frasi logiche composte da due frasi semplici che, quando espresse con variabili binarie diano luogo ad una tabella di verità di tipo EQ per la enumerazione delle combinazioni per cui la frase è vera
- P2.5 Ripetere i problemi P1.1-P1.4 per la enumerazione delle combinazioni per cui la frase è falsa
- P2.6 Esprimere la funzione Booleane per L del circuito di Fig. P2.1
- P2.7 Collegare la resistenza di Fig. 2.17 ai terminali Not(A) e B e determinare la funzione Booleane per L.
- P2.7 Collegare la resistenza di Fig. 2.17 ai terminali Not(B) e A e determinare la funzione Booleane per L.
- P2.9 Collegare la resistenza di Fig. 2.18 ai terminali Not(B) e A determinare la funzione Booleane per L.
- P2.10 Collegare la resistenza di Fig. 2.18 ai terminali Not(B) e Not(A) determinare la funzione Booleane per L.
- P2.11 Nella Fig. 2.20 Scambiare i terminali di massa con VDD e trovare le funzioni Booleane per la funzione F.

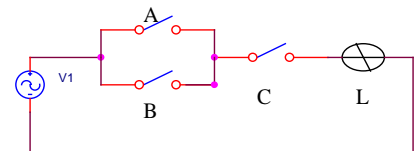


Fig. P2.1

CAPITOLO 2	21
LOGICA BINARIA	21
2.1 INTRODUZIONE ALLE VARIABILI BINARIE	21
2.1.1 <i>Frase Logiche: Vero o Falso</i>	21
2.1.2 <i>Assegnazione di un valore numerico a una variabile binaria</i>	27
Tab. 2.6 Tabella di verità AND con le due assegnazioni	27
Tab. 2.7 Tabella di verità OR con le due assegnazioni	27
2.1.3 <i>Rappresentazione grafica delle variabili binarie</i>	28
2.2 ELEMENTI DI CIRCUITI ELETTRICI CON CUI È POSSIBILE REALIZZARE FUNZIONI LOGICHE	31
2.2.1 <i>Uso dei Transistor CMOS come interruttori</i>	34
2.3. FAMIGLIE LOGICHE: GENERALITÀ	35
2.3.1 <i>Definizione dei livelli logici</i>	37
2.3.2 <i>Simboli standard per i circuiti elettronici che realizzano funzioni logiche</i>	39
2.3.3 <i>Fan_out</i>	40
2.3.4 <i>Rappresentazione grafica di segnali logici in funzione del tempo (TIMING)</i>	40
PROBLEMI.....	41
