

Istituto Professionale di Stato
per l'Industria e l'Artigianato
MORETTO
Via Luigi Apollonio, 21
BRESCIA

Anno Scolastico 1993-94

ESAME DI MATURITA' PROFESSIONALE PER
T.I.E.E.- Tecnico delle Industrie Elettriche ed Elettroniche

III Commissione

CIRCUITI COMBINATORI E NUMERICI

Motti Caprini Fabrizio

Classe 5CI TIEE

SOMMARIO

CIRCUITI COMBINATORI E NUMERICI.....	3
UN METODO DI PROGETTO	3
INDIVIDUAZIONE DELLE VARIABILI.....	4
STESURA DELLA TABELLA DELLA VERITÀ	5
TRADUZIONE DELLA TABELLA IN FUNZIONI BOOLEANE.....	7
SEMPLIFICAZIONE DELLE FUNZIONI	7
REALIZZAZIONE DI UNO SCHEMA A BLOCCHI.....	7
DECODIFICA.....	11
MATRICI DI DIODI.....	17
CONVERTITORI DI CODICE	19
CODIFICA	21
MULTIPLEXER (selettore).....	22
USO DEL MULTIPLEXER PER REALIZZARE RETI COMBINATORIE ..	25
DEMUTIPLEXER.....	26
DECODIFICATORE UTILIZZATO COME DEMULTIPLEXER.....	28
MULTIPLEXER UTILIZZATO COME GENERATORE DI FUNZIONI.....	28
DEMUTIPLEXER UTILIZZATO COME GENERATORE DI FUNZIONI	30
CIRCUITI NUMERICI.....	30
SOMMATORE	32
SOTTRATTORE.....	35
UNITÀ ARITMETICO-LOGICA (ALU).....	36

CIRCUITI COMBINATORI E NUMERICI

Il progetto e la costruzione di un sistema digitale si possono considerare in due differenti modi.

Un primo modo consiste nel considerare l'assegnazione di stati e la costruzione delle tabelle di verità, la minimizzazione, l'uso di porte logiche, ecc. Questo modo è di tipo *hardware*.

Il secondo modo consiste nell'affrontare i problemi cercando di scoprire una soluzione sistematica per la realizzazione del progetto ed il suo controllo. In poche parole, ci avviciniamo al progetto guardandolo nella sua totalità (*top*), per poi considerare via via i problemi di tipo costruttivo (*down*).

Questo metodo è conosciuto come *top-down* e viene utilizzato soprattutto per la realizzazione di sistemi digitali complessi come *microprocessori*, *mini computer*, ecc.

Per circuiti di tipo combinatorio, intendiamo quei blocchi funzionali in cui, avendo una o più variabili di ingresso I, le uscite Y sono funzione solamente dello stato attuale degli ingressi. Sono cioè quei circuiti, schematizzati qui in fondo che non hanno nessuna capacità di memorizzare nessuno stato precedente.



Schematizzazione di un circuito combinatorio.

Chiamiamo invece sequenziali quei circuiti la cui condizione dipende anche dagli stati precedenti delle uscite.

UN METODO DI PROGETTO

Quando si deve progettare un circuito digitale per il controllo di un processo industriale o per il trattamento delle informazioni sotto forma numerica, si può cercare di impostare il problema definendo alcune fasi di lavoro:

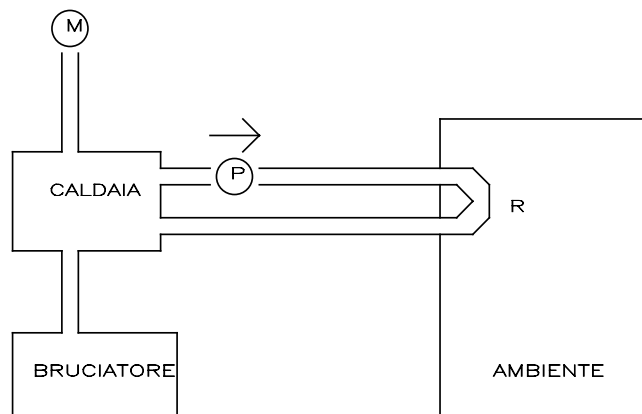
- 1) Individuazione delle variabili presenti nel sistema;
- 2) Stesura della tabella della verità del sistema completo;
- 3) Traduzione della tabella in funzioni Booleane;
- 4) Semplificazione delle funzioni;

- 5) Realizzazione di uno schema a blocchi funzionale;
- 6) Ulteriore semplificazione e minimizzazione.

Studieremo ora queste fasi con un esempio.

INDIVIDUAZIONE DELLE VARIABILI

Come esempio, prenderemo un sistema composto da una caldaia che debba riscaldare acqua per mantenere un ambiente ad una temperatura costante.



RISCALDAMENTO DI UN AMBIENTE:

M ≡ manometro
P ≡ pompa
R ≡ radiatore

Perché l'impianto funzioni correttamente è necessario fornirgli delle informazioni, tenendo presente che le variabili scelte e il loro stato logico sono state scelte in maniera casuale.

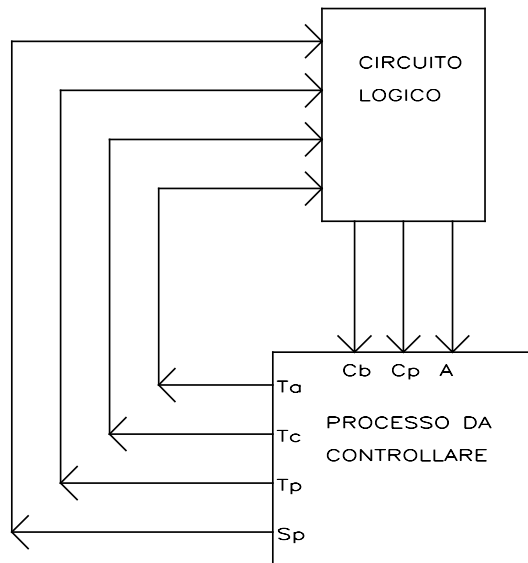
Un esempio di informazioni potrebbe essere:

- * la temperatura dell'ambiente che si vuole riscaldare (T_a), che può superare o no la soglia prefissata;
- * la temperatura dell'acqua presente nella caldaia (T_c), che non deve superare una soglia di 60 °C;
- * la temperatura dell'acqua oltre la quale la pompa di circolazione deve entrare in funzione, 30 °C;
- * un manometro di sicurezza che deve impedire alla pressione, in caso di guasto, di superare una data soglia di sicurezza.

Dall'elaborazione di questi dati forniti dal processo al circuito digitale, si devono avere dei segnali di uscita per il comando, tramite un'apposita interfaccia, dell'accensione del bruciatore (C_b), della pompa (C_p); e alla fine un segnale di allarme (A) che spenga tutto l'impianto e accenda una spia.

Lo schema a blocchi finale è rappresentato in figura.

Dopo aver definito le variabili d'ingresso e di uscita del sistema, associamo ad ognuna di queste un livello logico 0 oppure 1 che non potrà più essere cambiato in fase di progetto.



Schema a blocchi del controllo del controllo del processo

STESURA DELLA TABELLA DELLA VERITÀ

Se nella stesura della tabella della verità il numero delle variabili in ingresso è limitato (come nel nostro caso in cui ce ne sono quattro), si possono rappresentare tutte le loro possibili combinazioni (nel nostro caso $2^4 = 16$) anche se alcune di queste saranno differenti o impossibili da verificarsi.

Variabile	STATO DEL SEGNALE	Livello logico associato
T_a	ambiente al di sotto della temperatura ambiente al di sopra della temperatura	0 1
T_c	acqua al di sotto di 60 gradi acqua al di sopra di 60 gradi	0 1
T_p	acqua al di sopra di 30 gradi acqua al di sotto di 30 gradi	1 0
S_p	pressione al di sopra della soglia pressione al di sotto della soglia	1 0
C_b	il bruciatore deve essere spento il bruciatore deve essere acceso	0 1
C_p	la pompa deve essere spenta la pompa deve essere in funzione	0 1
A	allarme non in funzione allarme in funzione	0 1

Se invece il numero delle variabili è elevato e i casi impossibili od indifferenti non più trascurabili, risulta più agevole stendere una tabella delle verità contenente solo gli stati possibili, semplificando, così, notevolmente il lavoro di minimizzazione.

Nel nostro processo in esame si è supposto il seguente funzionamento: inizialmente l'ambiente che si vuol riscaldare è freddo, l'acqua nella caldaia è fredda e la pompa che la spinge in circolazione è ferma. Dopo l'accensione del bruciatore, la pompa entra in funzione non appena l'acqua supera i 30°C di temperatura e si spegne solo in caso di allarme o se la temperatura scende nuovamente al di sotto di 30°C.

Variabili in ingresso				Variabili in uscita			NOTE
Ta	Tc	Tp	Sp	Cb	Cp	A	
0	0	0	0	1	0	0	inizio funzionamento
0	0	0	1	0	0	1	processo fermo per allarme
0	0	1	0	1	1	0	l'acqua ha superato i 30 gradi di temp.
0	0	1	1	0	0	1	processo fermo per allarme
0	1	0	0	0	0	0	l'acqua ha superato i 60 gradi (impossibile)
0	1	0	1	0	0	1	fermo per allarme
0	1	1	0	0	1	0	e'accesa solo la pompa
0	1	1	1	0	0	1	fermo per allarme
1	0	0	0	0	0	0	l'ambiente ha raggiunto la temperatura
1	0	0	1	0	0	1	fermo per allarme
1	0	1	0	0	1	0	ambiente caldo e pompa in funzione
1	0	1	1	0	0	1	fermo per allarme
1	1	0	0	0	0	0	ambiente caldo, non scaldare
1	1	0	1	0	0	1	fermo per allarme
1	1	1	0	0	1	0	ambiente caldo e pompa in funzione
1	1	1	1	0	0	1	fermo per allarme

Il bruciatore si accende fintanto che l'ambiente è freddo e l'acqua nella caldaia è al di sotto di 60°C. Si spegne, invece, quando l'ambiente ha raggiunto la temperatura voluta o quando l'acqua sale al di sopra di 60°C. Nel caso di allarme, che si ha se la pressione supera un determinato valore, il manometro emette un segnale (S_p), che deve spegnere tutto l'impianto ed accendere una lampada spia.

TRADUZIONE DELLA TABELLA IN FUNZIONI BOOLEANE

Osservando la tabella della verità è evidente che, poiché il numero di 1 nelle variabili di uscita è minore del numero di 0, risulta più semplice scrivere le rispettive funzioni come somme di prodotti ottenendo:

$$C_b = \overline{T_a} \overline{T_c} \overline{T_p} \overline{S_p} + \overline{T_a} \overline{T_c} T_p \overline{S_p}$$

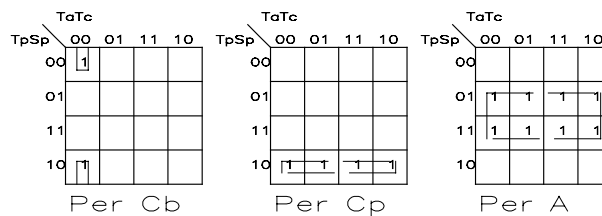
$$C_p = \overline{T_a} \overline{T_c} T_p \overline{S_p} + \overline{T_a} T_c T_p \overline{S_p} + T_a \overline{T_c} T_p \overline{S_p} + T_a T_c T_p \overline{S_p}$$

$$A = \overline{T_a} \overline{T_c} \overline{T_p} S_p + \overline{T_a} \overline{T_c} T_p S_p + \overline{T_a} T_c \overline{T_p} S_p + \overline{T_a} T_c T_p S_p + T_a \overline{T_c} \overline{T_p} S_p + T_a \overline{T_c} T_p S_p + T_a T_c \overline{T_p} S_p + T_a T_c T_p S_p$$

SEMPLIFICAZIONE DELLE FUNZIONI

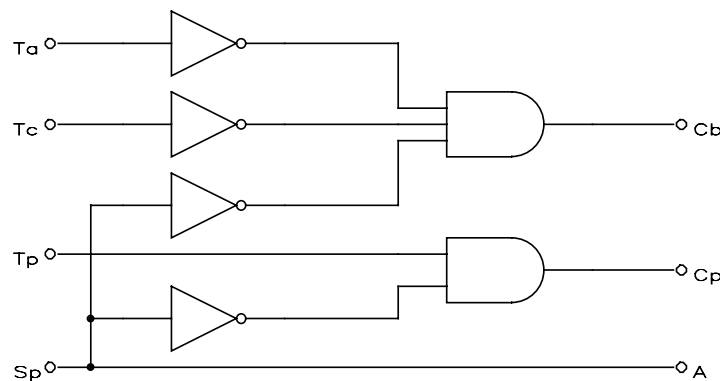
Lo scopo della semplificazione è quello di diminuire il numero delle porte necessarie, per rendere minimo l'ingombro del circuito, l'assorbimento di potenza, il costo e la complessità.

Tra tutti i metodi di semplificazione, abbiamo usato quello delle mappe di Karnaugh, che dà questi risultati:



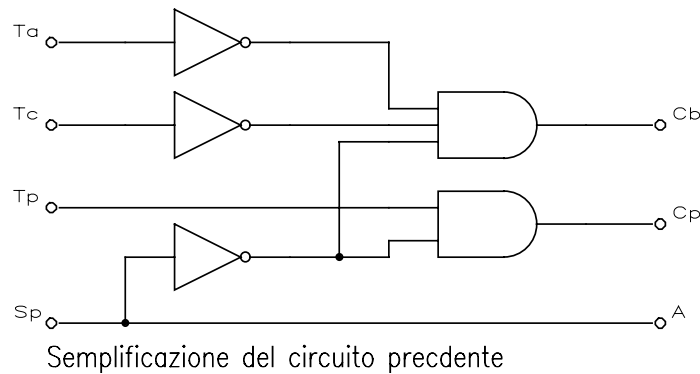
REALIZZAZIONE DI UNO SCHEMA A BLOCCHI

Dalle tre funzioni ottenute precedentemente, otteniamo il seguente schema:



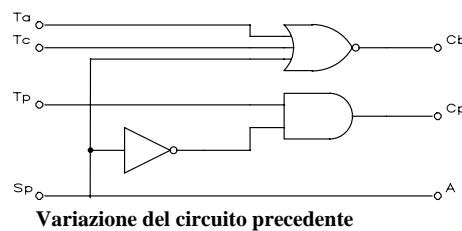
Questo circuito può essere ancora semplificato utilizzando vari metodi. Uno di questi è di usare anche parti di circuito realizzato da porte che sono usate per implementare un'altra funzione.

Per esempio, possiamo prelevare dal circuito sottostante la variabile $\overline{S_p}$ necessaria per formare C_b , dall'inverter utilizzato per ottenere la variabile C_p , come mostrato nella figura sottostante, risparmiando un inverter.



Un altro metodo per la semplificazione si può ottenere applicando le leggi di De-Morgan. In particolare la variabile d'uscita $C_b = \overline{T_a} \cdot \overline{T_c} \cdot \overline{S_p}$, può essere trasformata in

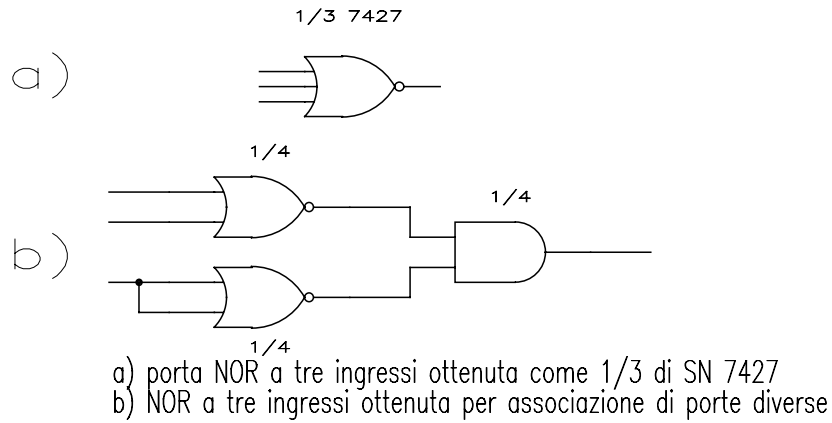
$$C_b = \overline{T_a + T_c + S_p},$$



Risulta facile capire che questo modo di operare risulta tanto più utile quanto più è complesso il circuito da realizzare.

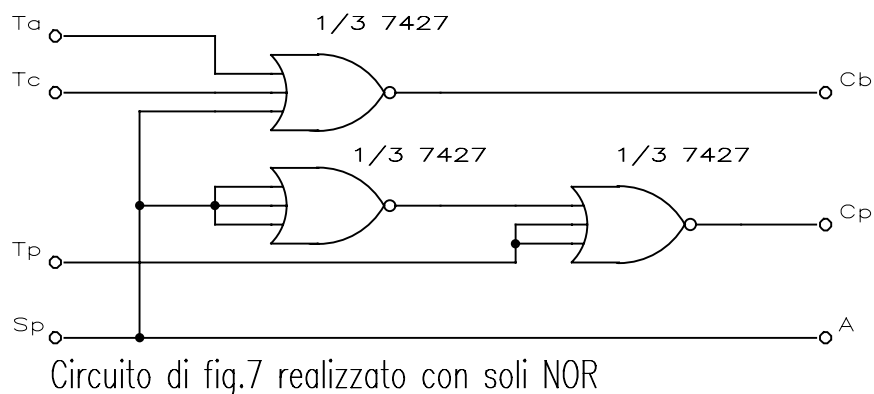
La porta NOR a tre ingressi può ottenersi o con una sola porta ad ingressi multipli, oppure associando tra di loro più porte con meno ingressi.

Sotto si possono vedere degli esempi di porte diverse.



Andando avanti in questa direzione e usando le leggi dell'algebra di commutazione, il circuito di partenza può ottenersi in un'estrema varietà di configurazioni più o meno valide a seconda delle necessità di progetto.

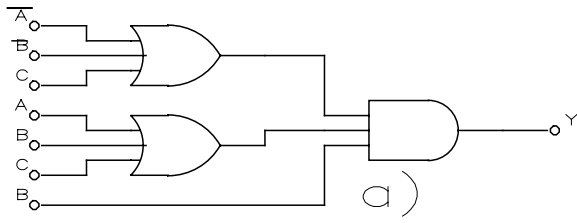
Bisogna tener presente che per ragioni d'ingombro, potenza e costo, è meglio utilizzare tutte le porte di un integrato. E' per questo motivo che il più delle volte si costruiscono circuiti che utilizzano solo porte NOR o NAND. Con questa soluzione si può costruire il nostro circuito nel seguente modo.



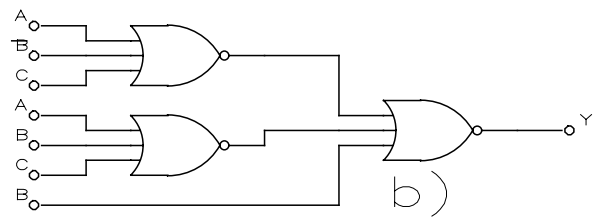
Per facilitare questa conversione si possono usare le seguenti regole:

1) Una funzione costituita da somma di prodotti, può essere convertita direttamente in una rete di NAND, e cioè tutte le porte AND ed OR sono sostituite da porte NAND. Se la somma di prodotti include una singola variabile connessa all'OR, nella conversione deve essere complementata.

2) Una funzione costituita da un prodotto di somme, può essere convertita direttamente in una rete di soli NOR, e cioè tutte le porte AND ed OR sono sostituite da porte NOR. Se il prodotto di somme comprende una singola variabile connessa all'AND, nella conversione deve essere complementata.



a) circuito combinatorio misto



b) realizzato con sole porte NOR

DECODIFICA

Una definizione generale di circuito digitale decodificatore coincide con quella data per definire i circuiti combinatori, appartengono cioè appartengono a questa categoria tutti quei sistemi che danno una determinata uscita per ogni diversa condizione degli ingressi.

Una semplice classificazione dell'utilizzo dei decodificatori può essere la seguente:

- a) passaggio da un sistema di numerazione ad un altro;
- b) indicazione dello stato delle variabili in ingresso;
- c) indicazione di parità e disparità;
- d) indicazione di maggioranza e minoranza;
- e) conversione di codici, ecc.

Anche in questo caso useremo un esempio per definire un metodo generale che consenta di costruire qualsiasi tipo di decodificatore.

Come esempio supponiamo di voler effettuare la conversione di un particolare codice BCD a quello decimale.

Il codice B.C.D. (Binary Coded Decimal), è un sistema in cui ogni cifra decimale, dallo 0 al 9, è rappresentata da quattro bit in notazione binaria, come mostrato nella tabella sottostante. In questo modo, un qualsiasi numero decimale di più cifre, può essere scritto nel codice B.C.D. come un insieme di gruppi di quattro bit ciascuno.

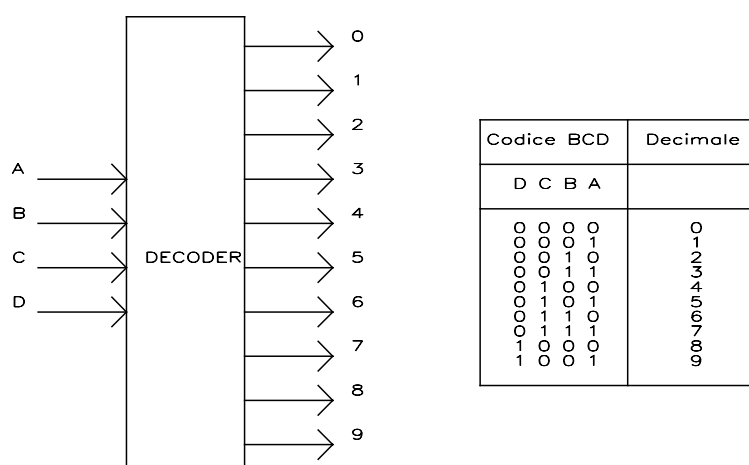


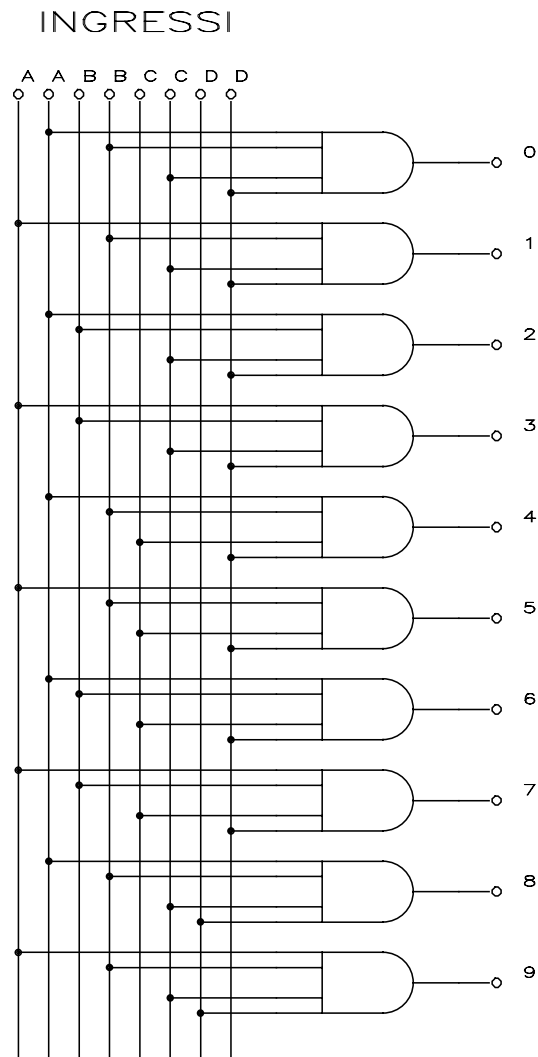
Tabella di conversione da BCD a decimale e schema a blocchi di un decoder

Dalla tabella di conversione, che è anche la tabella della verità per il decoder nella forma sotto riportata, si possono trarre i dieci mintermini rappresentanti le dieci cifre:

B.C.D				Decimale									
D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

Tabella della verita' per il decoder da BCD a decimale

In questo modo, non potendo fare nessuna semplificazione, il circuito di decodifica sarà costituito da dieci AND a quattro ingressi ciascuno, che è chiamato col nome di "matrice rettangolare".



Per ognuna delle combinazioni degli ingressi rappresentata nella tabella, una sola delle uscite si porterà al livello alto individuando il numero decimale corrispondente. Per semplicità, le variabili complementate degli ingressi sono state considerate come a disposizione senza ulteriori passaggi.

In realtà con quattro bit sono possibili $2^4 = 16$ possibili combinazioni diverse anziché 10. Queste altre sei combinazioni in più sono utilizzabili per la semplificazione del circuito se vengono considerate come condizione indifferente Φ .

La nuova tabella risulta allora indicata in figura.

A questo punto, per la semplificazione delle dieci funzioni Y che si vogliono ottenere, è necessario costruire le dieci mappe di Karnaugh per la semplificazione tenendo però conto, dove è possibile, delle condizioni indifferenti. Come esempio si può vedere che per la cifra 8 avremo:

	AB			
CD	00	01	11	10
00				
01	1	0	0	
11	0	0	0	0
10				

ottenendo così il mintermine $m_8 = \overline{A}D$ anziché $\overline{A}\overline{B}\overline{C}D$, e così via anche per gli altri.

Nello schema è stato inserito anche un ingresso di enable, che serve ad abilitare il decoder al funzionamento. Infatti, se per esempio l'ingresso di enable è al livello 0, tutte le uscite saranno portate al livello 0 per qualsiasi combinazione degli ingressi a livello alto.

Se, invece, l'ingresso di abilitazione è al livello 1, le uscite dipendono normalmente dagli ingressi. L'aggiunta di questo ulteriore comando, anziché complicare il circuito, lo rende più flessibile negli impieghi di interfacciamento.

Se anziché un decoder da BCD a decimale, fosse necessario un circuito con più di quattro ingressi per una diversa decodifica, aumenterebbe la complessità e l'ingombro del sistema realizzato con matrici rettangolari. Una prima semplificazione può ottenersi con una configurazione matriciale detta "ad albero".

Questa si può ottenere realizzando una logica di selezione per livelli successivi, scrivendo cioè le funzioni di uscita ottenute dalla tabella nel modo seguente:

$$Y_0 = \overline{ABCD} = \overline{[(\overline{AB})C]D}$$

$$Y_1 = \overline{A}BCD = \overline{[(\overline{A}B)C]D}$$

$$Y_2 = \overline{AB}CD = \overline{[(\overline{AB})C]D}$$

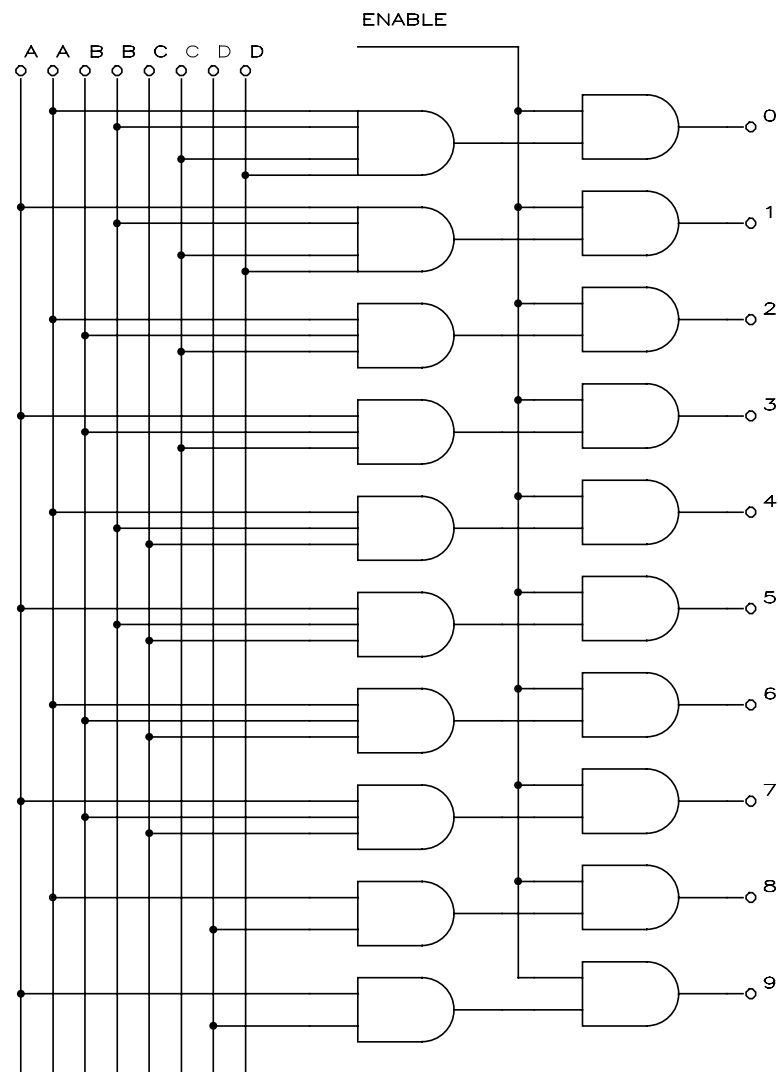
$$Y_3 = A\overline{BCD} = \overline{[(AB)\overline{C}]D}$$

.

.

.

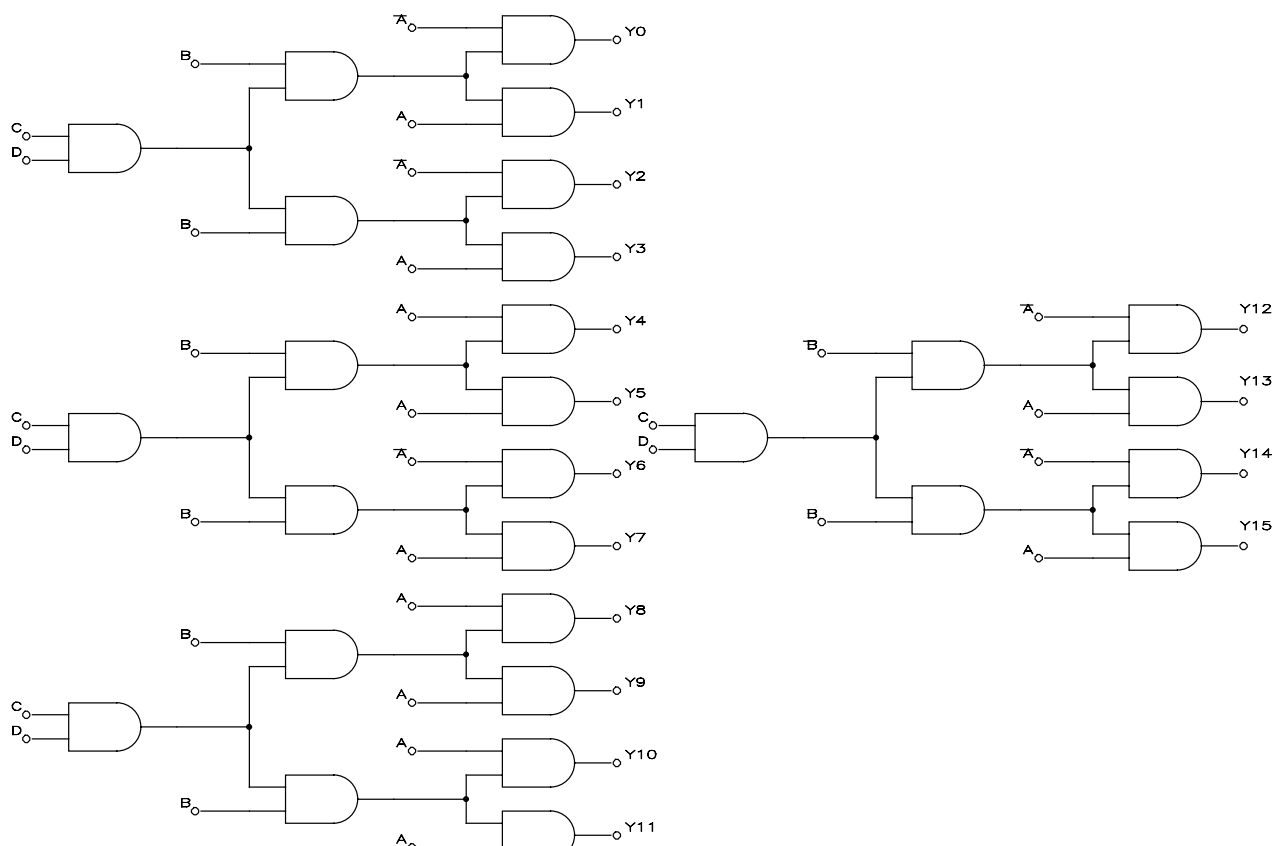
$$Y_{15} = ABCD = [(AB)C]D$$



Matrice di decodifica semplificata con ingresso di enable

Rispettando questi raggruppamenti, si ottiene lo schema circuitale mostrato nella figura sottostante. Più in generale, se anziché decodificare quattro ingressi occorre decodificare una parola di n bit, saranno necessarie un numero di porte AND a due ingressi date da:

$$N = \sum_{i=2}^n 2^i$$



Matrice di decodifica per livelli successivi (Matrice ad albero)

Se, invece, il raggruppamento delle equazioni d'uscita viene effettuato nel modo seguente:

si ottiene una costruzione detta "matrice a doppio albero", mostrata sotto, per il caso di quattro variabili (A, B, C, D) in ingresso.

$$Y_0 = (\overline{AB})(\overline{CD})$$

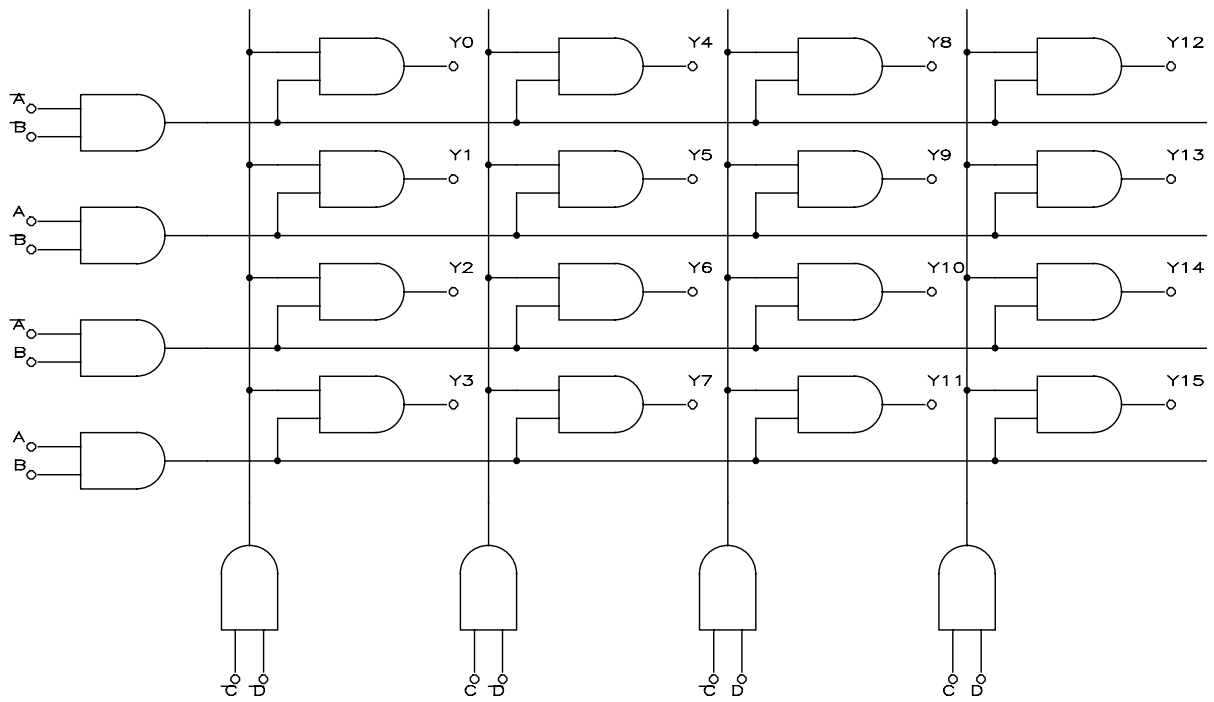
$$Y_1 = (\overline{AB})(\overline{CD})$$

$$Y_2 = (\overline{AB})(\overline{CD})$$

●●●●●●●●●●●●●●●●

$$Y_{15} = (AB)(CD)$$

Se il numero degli ingressi non è multiplo di 4, si perde la simmetria realizzativa, ma il numero di porte risparmiate con l'uso di questa configurazione è sensibile e cresce notevolmente al crescere della complessità.



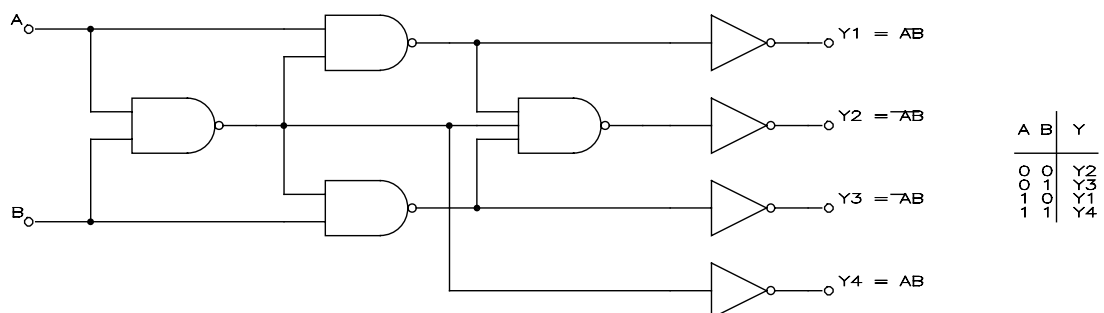
Matrice a doppio albero

In pratica esistono degli integrati MSI che svolgono questa stessa ed altre funzioni come, ad esempio, il decoder SN 7442 (four line to ten line decoder) della Texas Instruments. Il suo simbolo è rappresentato nella figura sottostante e come si può vedere dalla presenza dei cerchi sulle uscite, è attivo al livello basso, e cioè tutte le uscite sono a livello logico 1, tranne quella corrispondente alla configurazione degli ingressi che si porta al livello 0.

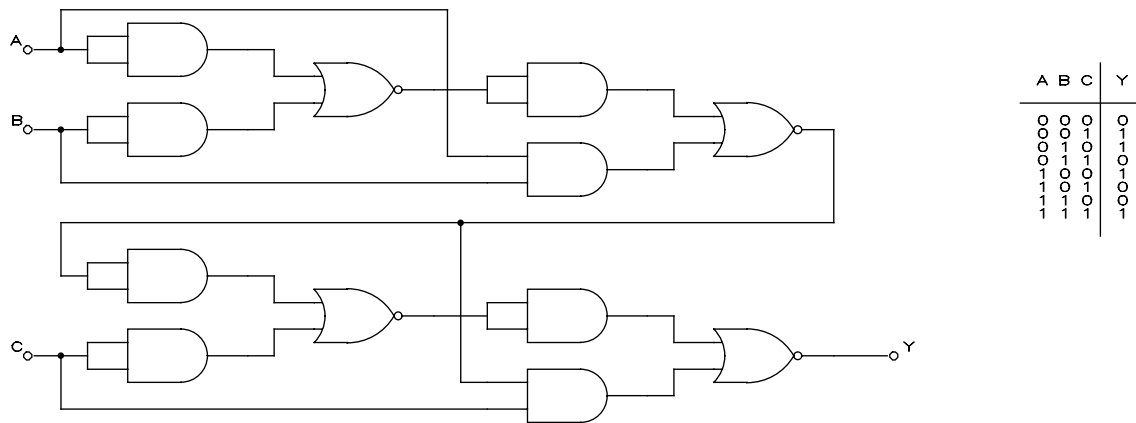
Nelle figure mostrate di seguito mostreremo gli schemi e le relative tabelle della verità di tre diversi decoder che sono rispettivamente:

- * rivelatore di combinazioni,
- * rivelatore di disparità,
- * rivelatore di maggioranza.

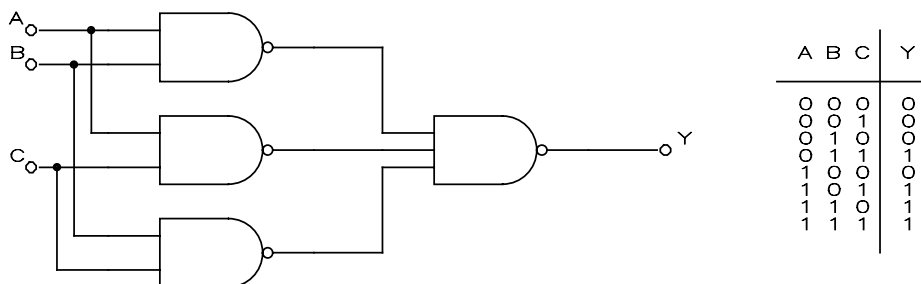
ciascuno dei quali può essere realizzato usando la stessa metodologia esposta precedentemente.



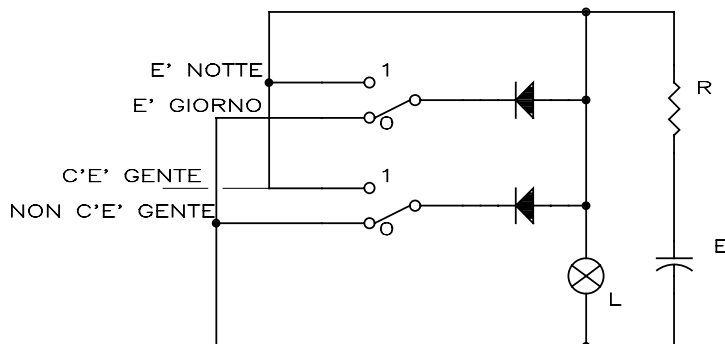
Ogni uscita indica lo stato delle variabili in ingresso (Texas Instruments)



L'uscita e' 1 solo quando e' dispari il numero degli ingressi di livello 1



L'uscita e' 1 quando la maggioranza degli ingressi e' al livello 1



La lampada L e' accesa solo se ambedue gli interruttori sono nella posizione 1

MATRICI DI DIODI

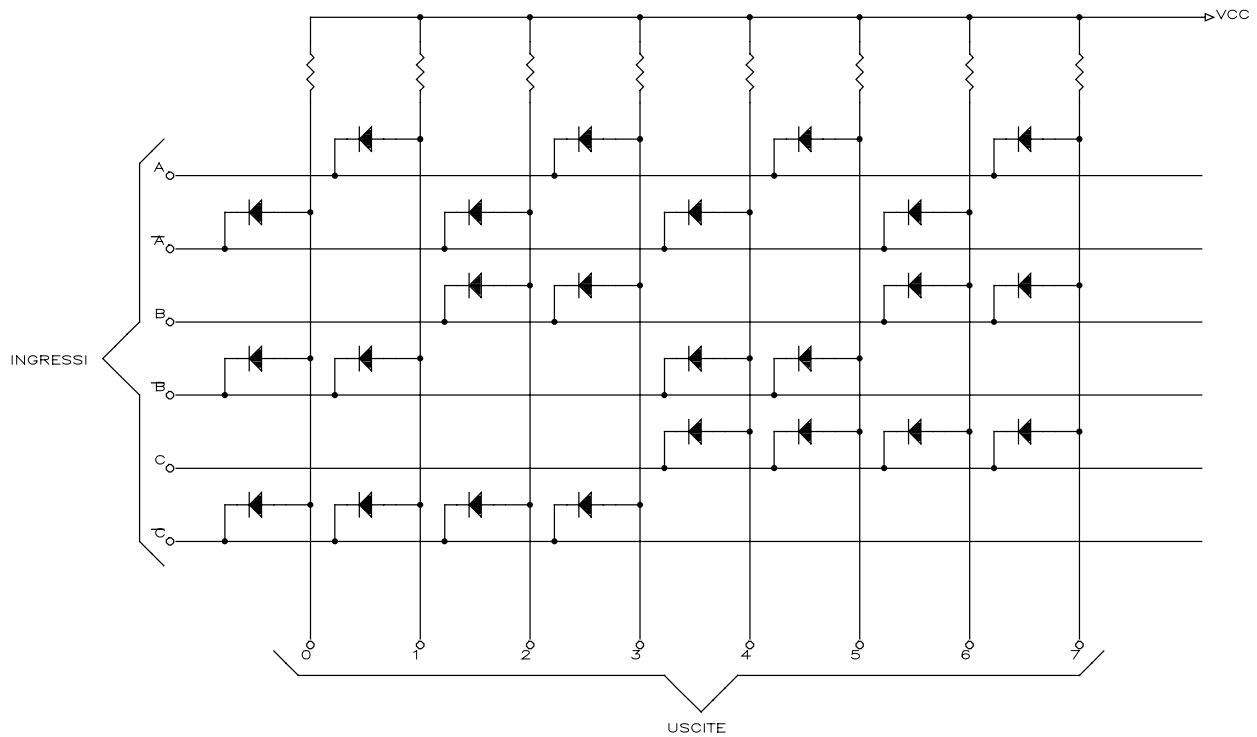
Si può realizzare la decodifica delle informazioni mediante di matrici di diodi. Se supponiamo di decidere l'accensione di una lampada in funzione di due fattori che potrebbero essere la presenza o assenza di persone e la luce diurna o notturna. La lampada si dovrebbe accendere solo se contemporaneamente sono verificate le condizioni che sia notte e che siano presenti delle persone.

Un semplice schema a matrice di diodi che svolga questa funzione è rappresentato nella figura sottostante.

La lampada L sarà accesa solo nel caso che ambedue gli interruttori, che forniscono le informazione in ingresso, siano nella posizione 1 (notte + gente);

in tutti gli altri casi la lampada rimarrà spenta poiché il suo morsetto positivo viene collegato alla massa, tramite almeno un diodo.

Nella figura qui rappresentata è mostrato un decoder da tre ingressi ad otto uscite e la sua relativa tabella della verità, realizzato con una matrice di diodi. La matrice è composta da tante righe orizzontali quanti sono gli ingressi (tre più tre complementati) e da tante colonne verticali quante sono le uscite.



INGRESSI			USCITE							
C	B	A	0	1	2	3	4	5	6	7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Matrice di decodifica da tre ad otto uscite e relativa tabella della verità'

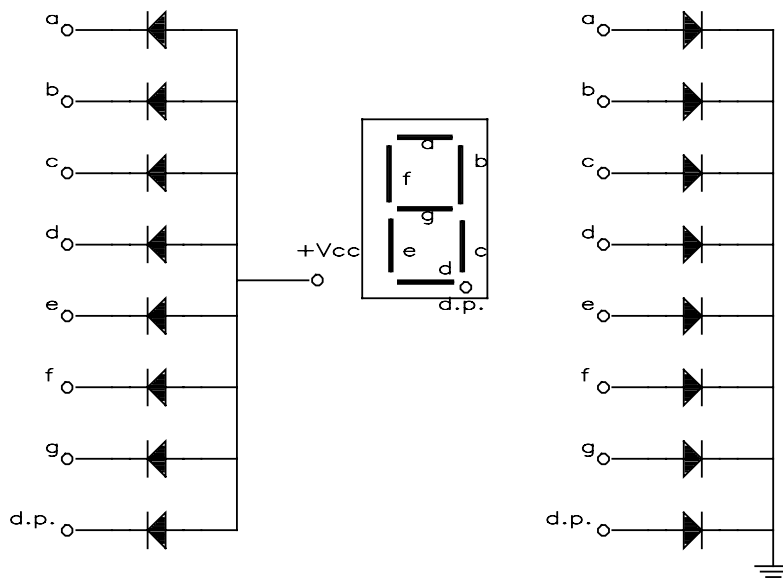
Ciascuna delle uscite andrà al livello 1 (alto) solo quando in ingresso comparirà la combinazione corrispondente. La costruzione di una matrice come questa è abbastanza semplice con l'aiuto della tabella della verità. Infatti per lo 0 si ha: $Y_0 = \overline{A}\overline{B}\overline{C}$, per cui se gli ingressi A, B e C sono al livello 0, i tre diodi

connessi ad $\overline{A}, \overline{B}$ e \overline{C} che si troveranno al livello 1 (+ Vcc), non condurranno facendo sì che l'uscita 0 si porterà al livello 1. Tutte le altre uscite hanno, invece, almeno un diodo che le collega al livello basso. In pratica avremo un 1 in uscita solo quando nessun diodo ad essa collegato è posto al livello zero.

Fra le tante applicazioni delle matrici a diodi, merita particolare attenzione la "griglia di programmazione". Essa è costituita da due serie di piste metalliche, una orizzontale e una verticale, isolate tra di loro e poste una sull'altra. Mediante spinotti contenenti diodi, possono essere messe in contatto agli incroci e realizzare in tal modo qualsiasi funzione di codifica e decodifica. Alcune tipiche applicazioni sono il controllo di macchine utensili, l'illuminazione di teatri, sistemi automatici di telefonia, programmazione apparati di collaudo, ecc.

CONVERTITORI DI CODICE

Per la visualizzazione delle lettere dell'alfabeto e dei numeri, generalmente si usano dei dispositivi detti displays alfanumerici. Quello più comunemente utilizzato per i soli numeri è il display a sette segmenti a diodi LED (Light Emitting Diode) oppure a cristalli liquidi LCD (Liquid Crystal Displays). Il display a sette segmenti è caratterizzato dal possedere sette ingressi indicati con a, b, c, d, e, f, g, (uno per ciascun segmento) più un eventuale ottavo ingresso d.p. (decimal point).



Display a sette segmenti ad anodo comune (a) e a catodo comune

Nella figura sottostante sono mostrate alcune cifre ottenute facendo illuminare in modo opportuno i segmenti. Ad esempio lo zero può essere visualizzato

accendendo tutti i segmenti ad esclusione di g; il numero 1 accendendo b e c; il numero 2 è ottenuto accendendo a, b, d, e, g, e così via per altre cifre.

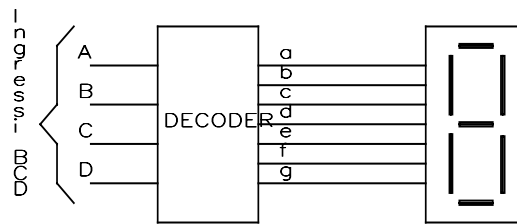
Esistono due diversi tipi di display a sette segmenti:

- 1) display ad anodo comune;
- 2) display a catodo comune.

Si usa il termine ad anodo comune quando tutti gli anodi dei diodi LED che formano i segmenti e il punto decimale sono collegati all'alimentazione. L'illuminazione di un segmento avviene soltanto quando il diodo LED corrispondente è in conduzione, situazione questa che si verifica quando il catodo è portato al livello logico basso.

Viceversa, un display è detto a catodo comune quando tutti i catodi dei diodi LED che formano i sette segmenti sono collegati a massa. Per fare in modo che un segmento si accenda occorre polarizzare direttamente l'anodo del diodo corrispondente portandolo al livello logico alto.

In figura è mostrato lo schema a blocchi che serve per un visualizzatore a catodo comune e la tabella della verità del decodificatore da BCD a sette segmenti.



Cifra decimale	Ingressi BCD				Uscita (segmenti)						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1

Schema a blocchi e tabella della verità per un display a sette segmenti a catodo comune con decoder ad ingressi B

La struttura interna del decodificatore può facilmente ricavarsi costruendo la mappa di Karnaugh per ciascuna delle sette funzioni di uscita a, b, c, d, e, f, g,

ottenendo le espressioni semplificate seguenti (consideriamo anche le condizioni di indifferenza):

$$a = AB + AC + \overline{AC} + D$$

$$b = AB + \overline{AB} + \overline{C}$$

$$c = A + \overline{B} + C$$

$$d = \overline{ABC} + \overline{AB} + \overline{AC} + \overline{BC}$$

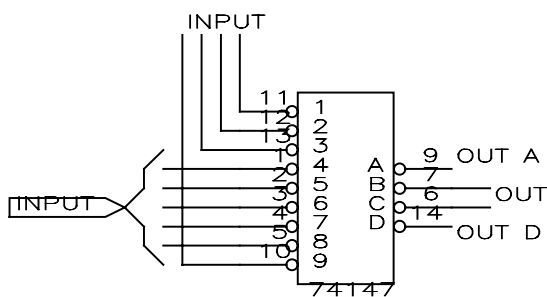
$$e = \overline{AB} + \overline{AC}$$

$$f = \overline{AB} + \overline{AC} + \overline{CB} + D$$

$$g = \overline{AB} + \overline{AC} + \overline{AC} + D$$

CODIFICA

I circuiti codificatori (encoder) svolgono la funzione opposta rispetto a quelli di decodifica e cioè quella di generare un codice da un set di informazioni poste all'ingresso. Questo genere di operazioni è meno frequente nel progetto di un sistema digitale, ma riveste anch'esso una certa importanza soprattutto per quanto riguarda i codificatori con priorità (priority encoders). Nella figura sottostante si può osservare il diagramma delle connessioni e la tabella della verità dell'encoder con priorità 74147 della Texas Instruments.



Ingressi									Uscite			
1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	L	L
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	L
L	H	H	H	H	H	H	H	H	H	H	H	L

Codificatore con priorità' 74147

H = livello alto
L = livello basso
X = indifferente

Questo integrato effettua la codifica da decimale a BCD (nove linee in ingresso e quattro in uscita).

Come si può vedere, non c'è la linea dello zero decimale poiché questa situazione è ottenuta quando tutti gli ingressi sono a livello alto. In questo particolare tipo di codificatore, i dati in ingresso e in uscita sono attivi al livello basso. Il funzionamento con la priorità fa sì che soltanto il dato di valore più

elevato posto all'ingresso venga codificato in uscita, mentre gli altri non vengono presi in considerazione e perciò sono indicati con X. Spesso in questi circuiti esiste un ingresso supplementare chiamato E1n (enable input) e due uscite chiamate E0 (enable output) e Gs (group signal).

Come si vede dalla seguente tabella della verità relativa all'encoder a otto bit con priorità 74148, se l'ingresso E1 è alto, tutte le uscite sono alte (si ricordi che il livello attivo è quello basso).

Se, invece, E1 è basso (L) e tutti gli ingressi sono bassi, l'uscita E0 va al livello basso. Infine, se E1 è basso (attivo) ed è attivo almeno un ingresso, anche Gs diviene attivo (L) ed E0 diventa alto.

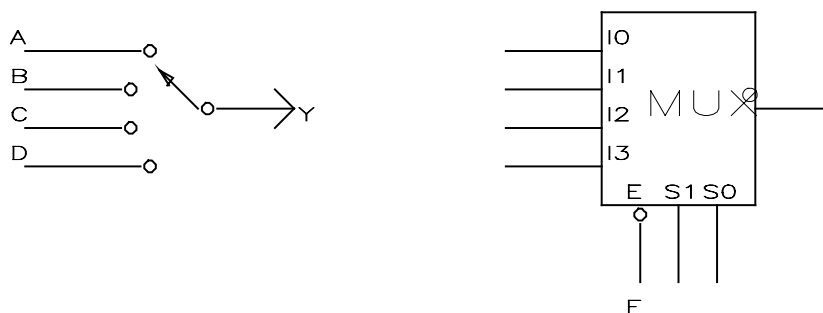
Le uscite E0 e Gs vengono utilizzate quando, ad esempio, si vuole realizzare un codificatore con priorità con più di otto ingressi.

INPUTS									OUTPUTS			
E ₁	0	1	2	3	4	5	6	7	A ₂	A ₁	A ₀	G _S E ₀
H	X	X	X	X	X	X	X	X	H	H	H	H H
L	H	H	H	H	H	H	H	H	H	H	H	H L
L	X	X	X	X	X	X	X	L	L	L	L	L H
L	X	X	X	X	X	X	L	H	L	L	H	L H
L	X	X	X	X	X	L	H	H	L	H	L	L H
L	X	X	X	X	L	H	H	H	L	H	H	L H
L	X	X	L	H	H	H	H	H	H	L	L	L H
L	X	X	L	H	H	H	H	H	H	L	H	L H
L	X	L	H	H	H	H	H	H	H	H	L	L H
L	L	H	H	H	H	H	H	H	H	H	H	L H

MULTIPLEXER (selettore)

Un multiplexer (MUX) è un selettore, con più ingressi di segnale e una sola uscita, capace di inviare a questa uscita il segnale presente all'ingresso selezionato.

Sotto viene schematizzato il funzionamento del MUX a quattro ingressi e la rappresentazione logica.



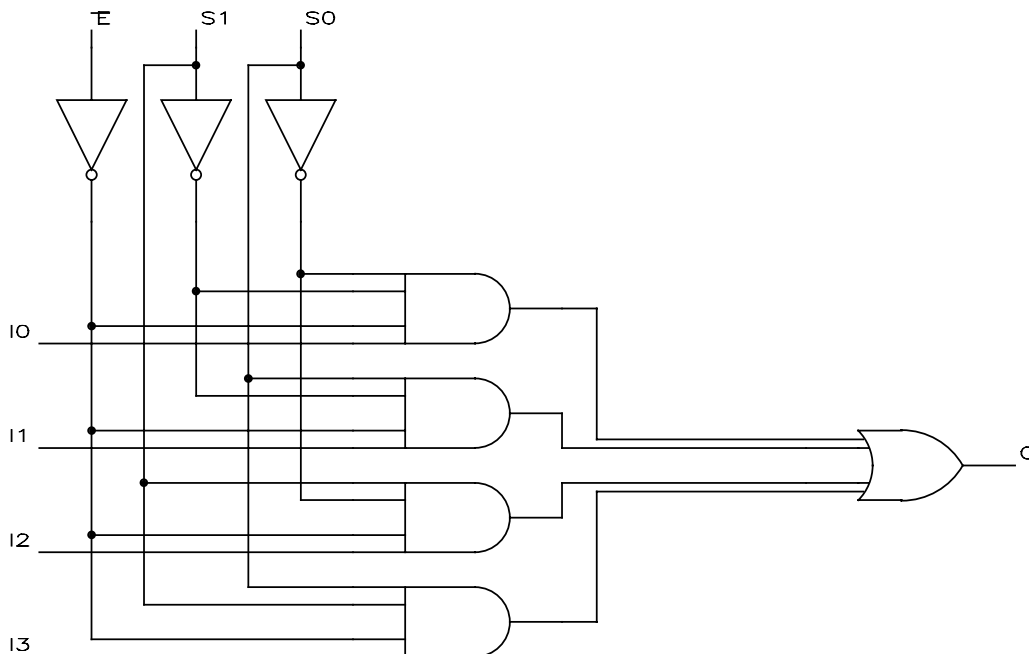
I0-I1-I2-I3 sono gli ingressi di segnale, S0-S1 quelli di selezione (l'ingresso selezionato è quello corrispondente al numero binario individuato da questi 2 bit), E è l'ingresso di enable (supposto, come in questo caso, attivo basso, con $\overline{E} = 0$ l'uscita assume il valore corrispondente all'ingresso selezionato, con $\overline{E} = 1$ l'uscita è il valore fisso =0).

Sotto viene riportata una possibile configurazione circuitale per il MUX.

La funzione realizzata risulta:

$$O = (I_3 S_1 S_0 + I_2 S_1 \overline{S_0} + I_1 \overline{S_1} S_0 + I_0 \overline{S_1} \overline{S_0}) E$$

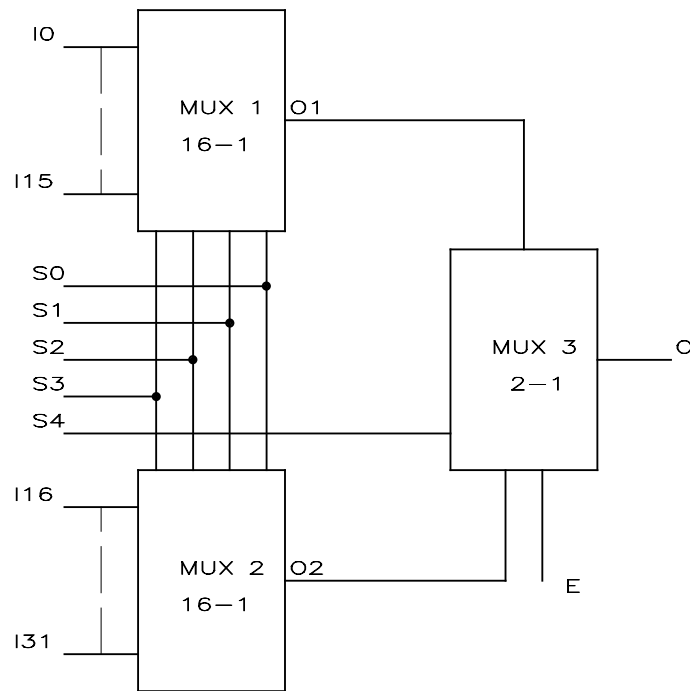
Come si vede se $\overline{E} = 0$ l'uscita corrispondente all'ingresso selezionato tramite S0 e S1.



Un esempio di MUX integrato a 16 ingressi di segnale è il 54/74150; Si noti che l'uscita fornisce il valore complementato degli ingressi, l'ingresso di enable è attivo basso e gli ingressi di selezione sono 4.

Controllando anche gli altri MUX si noti in particolare l'esistenza di enable che agiscono sull'uscita 3-state. Un esempio di questo genere si ha nel 4512B, che è un MUX a 8 ingressi di segnale, dove esiste un ingresso di disabilitazione 3-state che se posto a livello alto impone l'uscita ad alta impedenza, esiste però anche un ingresso di inibizione che se posto a livello alto impone l'uscita a livello basso (tra questi due ingressi è vincente quello 3-state).

Qualora non avessimo a disposizione un multiplexer con un sufficiente numero di ingressi di segnale si può applicare la tecnica mostrata nella figura sottostante.



Questo circuito si riferisce al caso di un MUX a 32 ingressi ottenibile da due 16 e uno a 2.

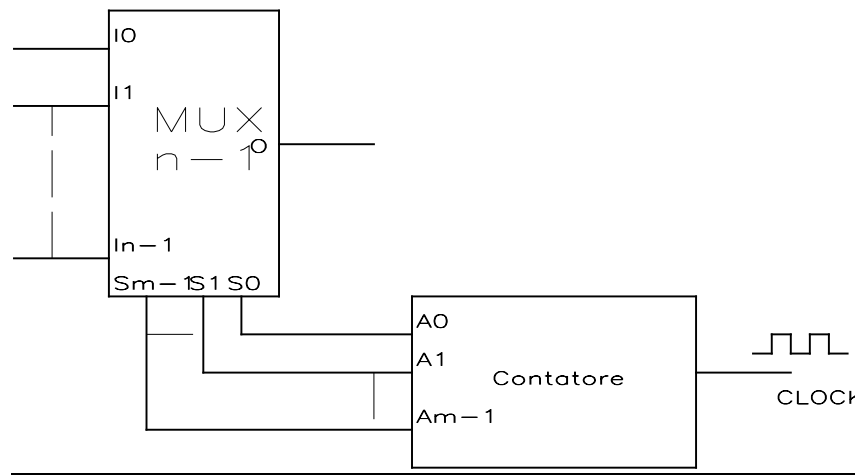
Si hanno in particolare 5 ingressi di selezione: i primi quattro S0, S1, S2, S3 selezionano contemporaneamente un ingresso dei MUX 1 e 2, l'ingresso S4 agisce sul MUX 3 selezionando O1 se vale 0 e O2 se vale 1.

Il multiplexer 3 è realizzabile con le normali porte logiche, ma è anche reperibile in forma integrata (es. 74/157 e 158).

In una trasmissione a distanza a più bit può risultare sconveniente trasmettere le singole parole a n bit usando n linee di trasmissioni distinte.

Una soluzione alternativa, che comunque rallenta la velocità di trasmissione dell'informazione, è quella di inviare su un'unica linea gli n bit in sequenza uno dopo l'altro.

Si effettua cioè una conversione parallelo serie del segnale binario, naturalmente in arrivo al sistema di trasmissione si deve poi procedere alla conversione inversa serie parallelo.



USO DEL MULTIPLEXER PER REALIZZARE RETI COMBINATORIE

Sebbene il multiplexer sia un circuito combinatorio che svolge una ben definita funzione logica è utilizzabile anche per realizzare una generica rete combinatoria.

Immaginiamo di dover, per esempio, realizzare la rete combinatoria sotto riportata.

Potremmo procedere alla minimizzazione, usando i criteri già noti, e arriveremmo a una soluzione con IC SSI.

In alternativa possiamo usare un MUX a 16 ingressi di segnale come riportato qui in figura.

Si usano come ingressi della rete combinatoria quelli di selezione e quelli di ingresso vengono collegati a 0 a 1 in relazione a cosa prevede la tabella della verità.

Come chiarimento di questa osservazione basta estendere al caso a 16 ingressi la funzione logica già considerata per il MUX precedente:

$$O = I_0 \bar{S}_3 \bar{S}_2 \bar{S}_1 \bar{S}_0 + I_1 \bar{S}_3 \bar{S}_2 \bar{S}_1 S_0 + I_2 \bar{S}_3 \bar{S}_2 S_1 \bar{S}_0 + \dots + I_{15} S_3 S_2 S_1 S_0$$

Come si vede, sono i valori 0 o 1 assegnati agli ingressi I a definire, per ogni combinazione $S_3 S_2 S_1 S_0$, il valore in uscita.

Un'altra soluzione, facente uso del MUX, è la seguente.

Si realizza la mappa corrispondente alla tabella della verità e la si divide in sottomappe, in modo che ognuna di queste, passando da una casella all'altra, cambi solo una variabile (nell'esempio cambia solo D).

Si realizza poi una tabella della verità considerando le uscite, corrispondenti alle singole sottomappe, rispetto alle variabili in ingresso che variano.

Si noti in particolare che qualora l'uscita vari, passando da una casella all'altra della stessa sottomappa, la si interpreta in funzione della variabile eliminata (nel nostro caso D).

Da questa tabella è facile ricavare il circuito finale che richiede in questo caso l'uso di un MUX a 8 ingressi di segnale e di un NOT.

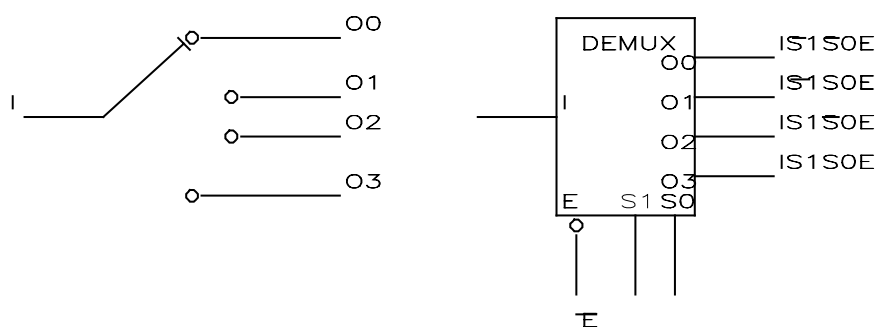
Qualora il NOT sia recuperabile da un integrato parzialmente utilizzato, questa soluzione risulta migliore della precedente.

Effettuando sottomappe di quattro caselle l'una è possibile usare un MUX a 4 ingressi di segnale, ma con l'aggiunta di altre porte esterne anche questa soluzione può risultare conveniente qualora le porte siano reperibili da integrati parzialmente utilizzati.

DEMULTIPLEXER

Un demultiplexer (DEMUX) svolge una funzione logica inversa del multiplexer, cioè è un circuito capace di inviare segnali provenienti da un'unica sorgente a più destinazioni.

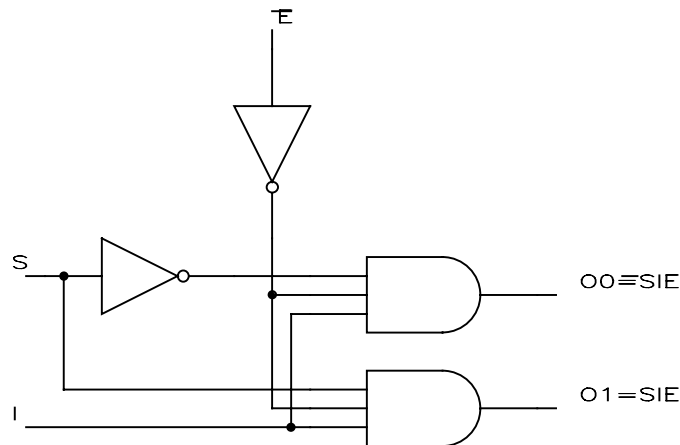
Qui sotto è mostrata un'analogia meccanica in cui, variando la posizione del deviatore, si seleziona una fra le varie uscite disponibili. Usando insieme un selettore e un distributore si può fare in modo che una sola linea possa servire alla trasmissione di più segnali diversi.



L'unico inconveniente è che i segnali devono essere inviati uno alla volta con una successione prestabilita.

Naturalmente la scelta dell'uscita avviene tramite m ingressi di selezione con $2^m = n$.

Qui sotto è schematizzato il comportamento del DEMUX ed è riportata la sua schematizzazione logica, nel caso di quattro uscite di segnali ed enable attivo basso.



Successivamente, invece, è riportata una possibile soluzione circuitale nel caso di due uscite.

Un esempio di demultiplexer è il 54/74154. Dalla sua tabella della verità si ricava che, oltre alle 16 uscite, sono presenti quattro ingressi di selezione A-B-C-D e due ingressi G1 e G2, che si possono considerare come degli enable attivi bassi. Se uno di questi o entrambi sono alti le uscite sono tutte alte, se entrambi sono bassi si ha l'uscita selezionata bassa e tutte le altre alte. In pratica G1 e G2, indifferentemente uno dall'altro, vengono scelti uno come entrata e l'altro come enable; se ad esempio si usa G1 come entrata e G2 come enable, posto $G2 = L$ l'uscita selezionata assume il valore di G1.

Anche in questo caso, così come nel paragrafo precedente, l'interruttore meccanico può essere sostituito da un dispositivo elettronico digitale che risulta più veloce, meno ingombrante e più affidabile.

Analogamente al MUX, che usava una codifica binaria per selezionare una fra le varie uscite. Il numero delle uscite direttamente selezionabili dipende dal numero degli ingressi di selezione o di indirizzo ed infatti nell'SN 74154 che ha quattro ingressi di indirizzo, si può distribuire l'informazione ad una qualsiasi delle sedici diverse uscite.

In relazione alle applicazioni dei decodificatori, selettori e distributori è utile porre in evidenza che:

- a) qualsiasi decodificatore può essere utilizzato come demultiplexer;
- b) qualsiasi demultiplexer può essere utilizzato come generatore di funzioni;
- c) qualsiasi demultiplexer può essere utilizzato per generare forme canoniche di somme e prodotti.

B.C.D				Decimale									
D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

DECODIFICATORE UTILIZZATO COME DEMULTIPLEXER

Supponiamo di voler utilizzare come distributore un decodificatore da BCD a decimale per il quale i legami tra i quattro ingressi (D, C, B, A) e le dieci uscite (Y0, ..., Y9) sono mostrati nella tabella sotto riportata.

Per ottenere ciò è necessario assegnare il dato da distribuire ad uno dei quattro ingressi (ad es. D) ed utilizzare i rimanenti tre (C, B, A) come ingressi di selezione, ottenendo in tal modo un demux ad otto uscite. Riscrivendo la tabella nel modo riportato qui sotto, si verifica facilmente come ciascuna delle prime otto uscite selezionate dagli ingressi C, B, A vada bassa (L) o alta (H) a seconda che il dato D sia basso o alto.

ing. dati	ing. di sel.			uscite selezionate							
D	C	B	A	0	1	2	3	4	5	6	7
L	L	L	L	L	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H
L	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	H	H	L	H	H	H	H
L	H	L	L	H	H	H	H	L	H	H	H
L	H	L	H	H	H	H	H	H	L	H	H
L	H	H	L	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	L
H	L	L	L	H	H	H	H	H	H	H	H
H	L	L	H	H	H	H	H	H	H	H	H
H	L	H	L	H	H	H	H	H	H	H	H
H	L	H	H	H	H	H	H	H	H	H	H
H	H	H	H	H	H	H	H	H	H	H	H

MULTIPLEXER UTILIZZATO COME GENERATORE DI FUNZIONI

Quasi tutte le funzioni logiche possono essere realizzate per mezzo di un multiplexer (selettore). Per ottenerle è necessario utilizzare un selettore capace di soddisfare tutti i mintermini della funzione, direttamente o mediante un

condizionamento, ed eliminando quelli non utilizzati. Un mintermine è condizionato dalla applicazione, di un livello logico appropriato, al corrispondente ingresso; mentre è eliminato applicando uno 0 all'appropriato ingresso dati. Se sono richiesti più mintermini di quelli a disposizione, essi possono essere creati dall'appropriata combinazione di uno o più dati in ingresso e/o usando più di un selettore. L'uso di n selectori con m ingressi dati forniscono un totale di $n \cdot m$ termini.

Come esempio si supponga di voler realizzare la funzione:

$$Y = \overline{A}CD + A\overline{B} + \overline{C}\overline{D}$$

usando il mux SN 74150. In tal caso occorre connettere un inverter all'uscita W (poichè è complementata), porre a livello zero l'ingresso di strobe per ottenere l'abilitazione ed applicare un 1 o uno 0 agli appropriati ingressi dei dati (E_0, \dots, E_{15}) per eliminare o condizionare i mintermini.

Sono simili parecchi approcci per determinare quali mintermini debbono essere eliminati o condizionati. Uno di essi è quello delle mappe di Karnaugh.

Quando una espressione logica è rappresentata mediante una di tali mappe, ciascuna cella rappresenta un mintermine e, nel caso di un selettore, il valore numerico ad essa assegnato corrisponde al numero dell'ingresso dati che governa il mintermine rappresentato dalla cella.

Nel caso della funzione $Y = \overline{A}CD + A\overline{B} + \overline{C}\overline{D}$ la mappa di Karnaugh è mostrata nella figura sottostante dove, ad esempio, la cella in alto a destra rappresenta l'ingresso E_1 ed il mintermine $\overline{A}\overline{B}\overline{C}\overline{D}$ mentre quella in basso a sinistra rappresenta l'ingresso E_4 ed il mintermine $\overline{A}BC\overline{D}$.

	AB			
CD \	00	01	11	10
00	1	1	1	1
01				1
11	1	1		1
10				1

Comparando la funzione data con quella dell'SN 74150:

$$\overline{S}(\overline{ABCDE_0} + \overline{ABCDE_1} + \overline{ABCDE_2} + \dots + ABCDE_{15})$$

si deduce che i mintermini controllati da $E_0, E_1, E_2, E_3, E_5, E_9, E_{12}, E_{13}, E_{14}$ sono necessari per cui i rispettivi ingressi devono essere posti al livello logico 1,

mentre i mintermini restanti non sono necessari per cui i corrispondenti ingressi $E_4, E_6, E_7, E_8, E_{10}, E_{11}, E_{15}$ devono essere connessi al livello logico zero.

L'utilizzo di selettori aventi tanti ingressi di indirizzo quanti sono le variabili della funzione di quattro variabili implementata precedentemente, anzichè l'SN 74150 può utilizzarsi un mux a tre o a due soli ingressi di indirizzo.

DEMULTIPLEXER UTILIZZATO COME GENERATORE DI FUNZIONI

La possibilità, per un demux, di essere impiegato per generare funzioni può essere facilmente rivelata scrivendone le equazioni di uscita. Per un demux a quattro uscite si ottiene:

$$Y_0 = \bar{P}_0 \bar{P}_1 A$$

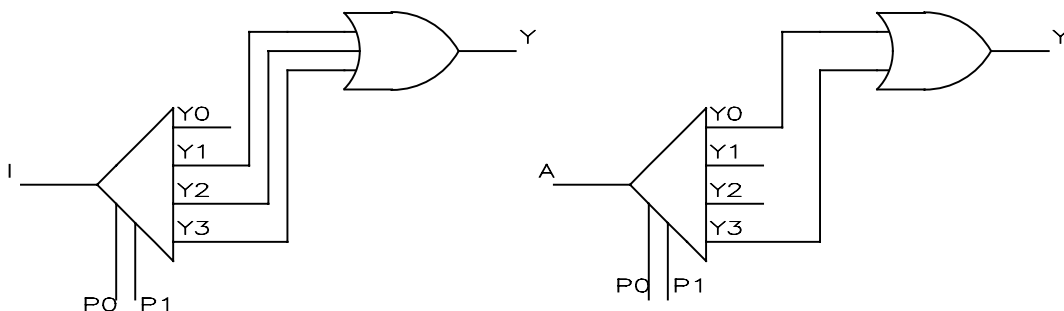
$$Y_1 = P_0 \bar{P}_1 A$$

$$Y_2 = \bar{P}_0 P_1 A$$

$$Y_3 = P_0 P_1 A$$

dove con P sono stati indicati gli ingressi di selezione e con A l'ingresso del dato da distribuire. L'uso di una porta OR per ottenere la somma logica delle uscite utili permette di realizzare una forma canonica della somma.

Le due figure sottostanti mostrano l'uso di un demux a quattro uscite per realizzare le funzioni $Y = P_0 \bar{P}_1 + \bar{P}_0 P_1 + P_0 P_1$ ed $Y = \bar{P}_0 \bar{P}_1 A + P_0 P_1 A$ rispettivamente.



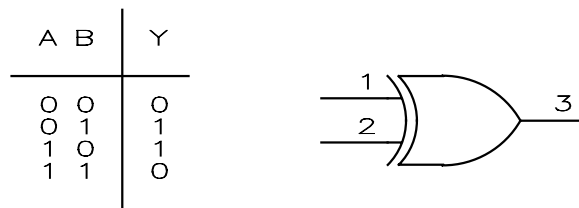
CIRCUITI NUMERICI

Una buona parte dei circuiti combinatori è costituita da circuiti aritmetici binari. Molto spesso queste operazioni sono ridotte più semplicemente a:

addizione, addizione e complementazione, addizioni ripetute con scorrimento di dati, rispettivamente.

In certi casi, dove la parte aritmetica è abbastanza importante, risulta più semplice usare integrati MSI-LSI detti unità aritmetico-logiche o ALU (Arithmetic Logic Unit), che possono eseguire sia operazioni logiche che aritmetiche su due parole, generalmente di 4 bit ciascuna, contemporaneamente.

Per poter capire meglio la realizzazione dei circuiti sommatore e sottrattori, è importante ricordare la funzione detta OR- esclusivo od anche operatore aritmetico. Sotto vengono riportate la tabella della verità ed il simbolo.

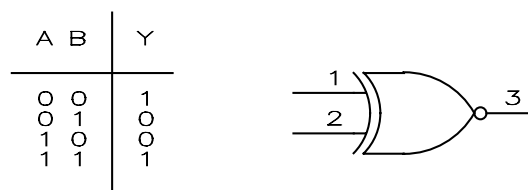


Questo speciale tipo di OR ha solo due ingressi ed una uscita, che è a livello alto solo se un ingresso è a livello alto e non entrambi (per questo viene chiamato circuito di anticoincidenza).

La funzione logica Or-esclusivo viene indicata con il simbolo \oplus e così si può scrivere:

$$Y = A \oplus B$$

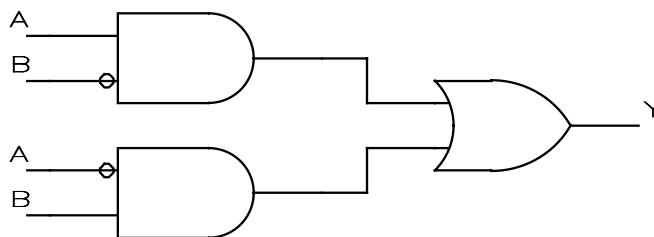
Sotto, invece, viene riportata la tabella della verità ed il simbolo logico dell'exclusive-NOR, che opera allo stesso modo dell'exclusive-OR ma l'uscita risulta invertita.



Come si vede, l'uscita è 1 solo quando gli ingressi sono entrambi allo stesso livello e per questo viene anche chiamato circuito di coincidenza. Con riferimento alla figura dell'EXOR, la funzione di uscita espressa come somma di mintermini, si può scrivere:

$$Y = \bar{A}B + A\bar{B}$$

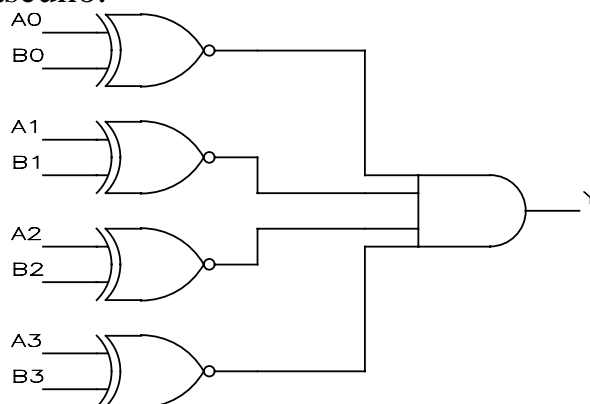
dalla quale si può trarre il seguente schema circuitale per l'Or-esclusivo.



Fra le altre applicazioni di questa porta logica, c'è l'utilizzo come comparatore. Infatti, se si vuole controllare se due linee è presente contemporaneamente lo stesso segnale, o se due macchine svolgono la stessa funzione, collegandole ad un OR- esclusivo si otterrà in uscita un 1 ogni volta che i due segnali non coincidono.

Inversamente, utilizzando un NOR-esclusivo, si può ottenere in uscita un 1 solo quando i due ingressi coincidono.

Se c'è il bisogno di verificare l'uguaglianza di due numeri di più bit ciascuno, occorrono tante porte NOR-esclusivo quanti sono i bit, per confrontare a due a due le cifre dello stesso ordine; le loro uscite vengono quindi collegate ad una porta AND a più ingressi, come è illustrato, per il caso del confronto di due numeri di quattro bit ciascuno.



SOMMATORE

Nelle tabella seguente sono riportate tutte le possibilità che si possono ottenere sommando due numeri A e B, di un bit ciascuno. Con S è stato indicato il bit di somma e con C (Carry) l'eventuale riporto.

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

Come si vede, la funzione somma coincide con la funzione OR- esclusivo, mentre quella di riporto con la funzione AND. Possiamo così scrivere:

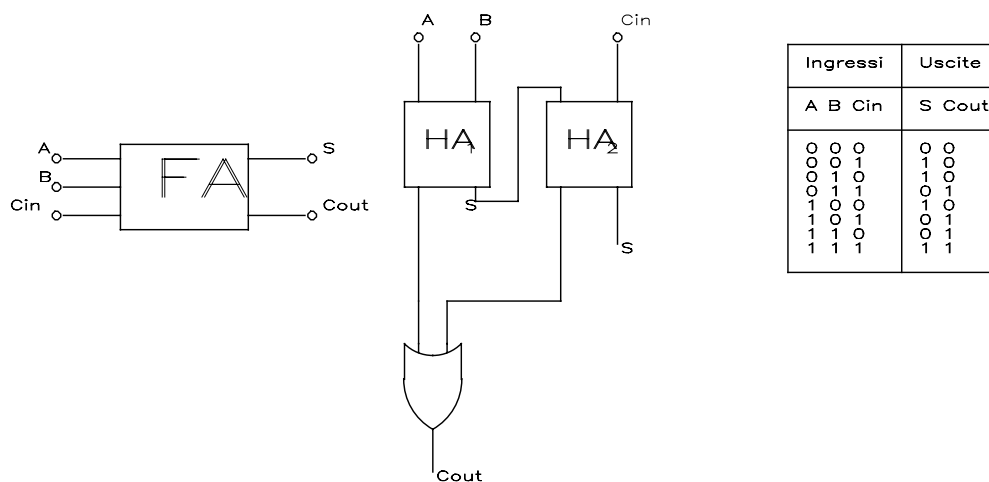
$$S = A \oplus B$$

$$C = A * B$$

Sotto viene mostrato lo schema ed il simbolo logico del circuito che realizza le precedenti operazioni e che viene chiamato semi-sommatore (Half-Adder). Questo nome deriva dal fatto che per ottenere una somma completa (Full-Adder) fra due numeri di più cifre, oltre ai bit dello stesso ordine occorre sommare anche il riporto eventualmente ottenuto dai due bit di ordine immediatamente inferiore.

In questo caso il circuito di full-adder si presenta con tre ingressi e due uscite. I tre ingressi sono costituiti da due bit A_n e B_n da sommarsi e dal riporto C_{IN} eventualmente ottenuto dalla somma dei due bit A_{n-1} e B_{n-1} . Le due uscite sono composte dal bit di somma dei tre ingressi e dall'eventuale riporto C_{OUT} da inviare al full-adder successivo.

Sotto viene mostrato il simbolo logico, la tabella della verità e lo schema circuitale di un full-adder.



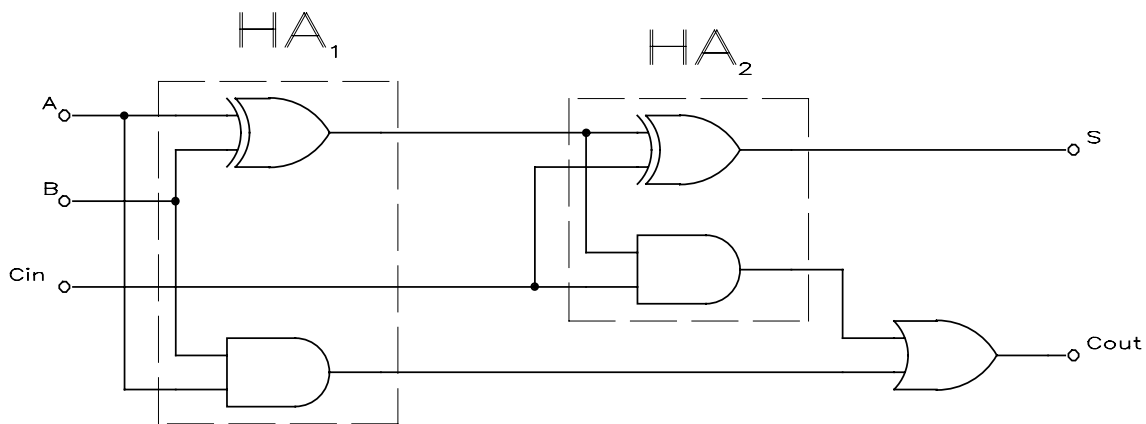
Questo circuito si può ricavare dalla tabella della verità, infatti, la funzione della somma si può scrivere:

$$\begin{aligned}
S &= \overline{A}\overline{B}C_{IN} + \overline{A}B\overline{C}_{IN} + A\overline{B}\overline{C}_{IN} + ABC_{IN} = \\
&= C_{IN}(\overline{A}\overline{B} + AB) + \overline{C}_{IN}(\overline{A}B + A\overline{B}) = \\
&= C_{IN}(A \oplus B) + \overline{C}_{IN}(A \oplus B) = (A \oplus B) \oplus C_{IN}
\end{aligned}$$

mentre per la funzione riporto:

$$\begin{aligned}
C_{OUT} &= \overline{A}BC_{IN} + A\overline{B}C_{IN} + AB\overline{C}_{IN} + ABC_{IN} = \\
&= C_{IN}(\overline{A}B + A\overline{B}) + AB(\overline{C}_{IN} + C_{IN}) = C_{IN}(A \oplus B) + AB
\end{aligned}$$

dalle quali si può trarre il seguente circuito.



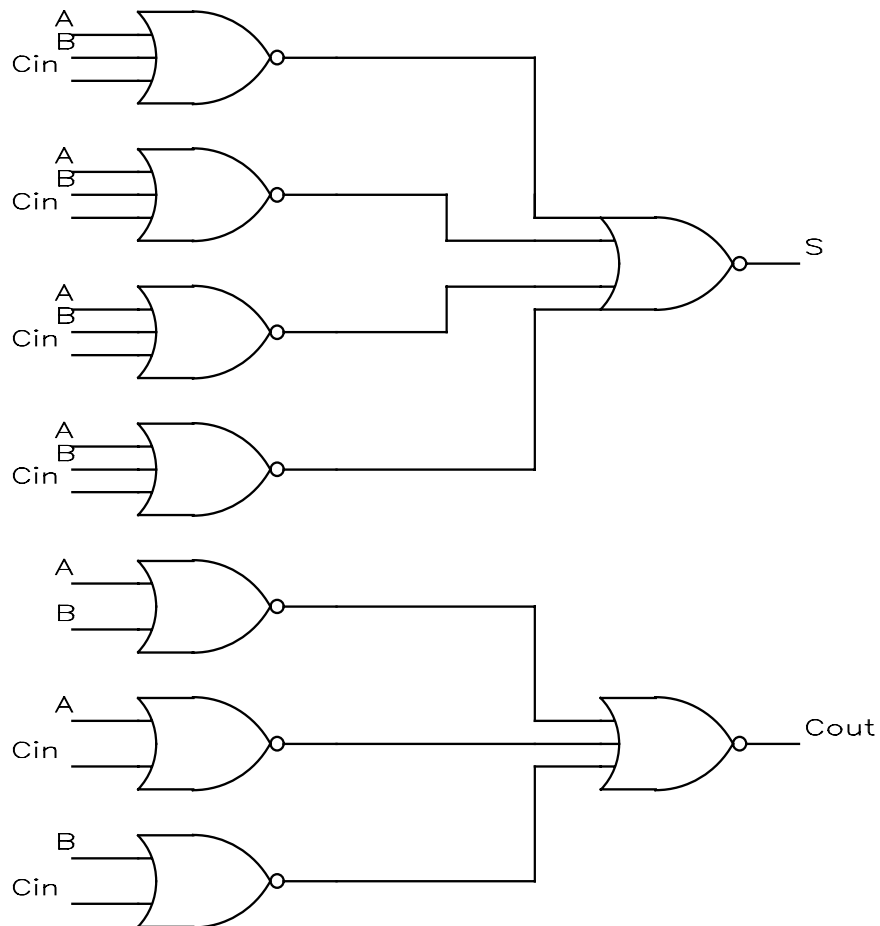
Anche in questo caso, questa rete combinatoria può essere realizzata con soli NOR o con soli NAND, partendo dall'equazione canonica espressa come somma di prodotti o prodotto di somme.

Sotto, per esempio, viene indicato lo schema di un full-adder realizzato con sole porte NOR.

I circuiti spiegati fino a qui possono essere definiti come semi- sommatore elementari, in grado cioè di sommare due bit più l'eventuale riporto.

Quando si vogliono sommare numeri di più bit ciascuno, il metodo più semplice è quello di realizzare un sommatore parallelo o asincrono, come quello sotto riportato, mettendo in cascata tanti F-A(Full-Adder) quanti sono i bit di ciascun numero binario da sommare.

In questo caso i quattro bit di ciascun numero vengono presentati contemporaneamente agl'ingressi dei full-adder e viene effettuata la somma se non è presente alcun riporto. Nel caso peggiore, invece, in cui ad ogni somma corrisponde un riporto, si avrà che questo bit traslerà dal primo H-A a tutti gli altri F-A fino all'ultimo.



Il tempo necessario per effettuare la somma, in quest'ultimo caso, è pari ad $N T_p$, dove N indica il numero di bit da sommarsi e T_p il tempo di propagazione di ciascun F-A. Sotto viene mostrato lo schema completo di un addizionatore da 4 bit, che usa solo porte elementari ed in cui i numeri da sommare sono individuati dalle posizioni degli otto deviatori.

Molto spesso, per eliminare l'inconveniente della lentezza dell'addizionatore parallelo, si fa ricorso ad uno speciale circuito combinatorio detto "generatore simultaneo di riporto", per mezzo del quale prima si genera il riporto e poi viene effettuata l'addizione. In questo modo il tempo di ritardo è fissato e valido per qualsiasi somma. In altri casi, invece, si usano addizionatori seriali o sincroni che operano in concomitanza con un segnale esterno di clock.

SOTTRATTORE

Nella maggior parte della apparecchiature per il trattamento automatico delle informazioni, l'operazione aritmetica della sottrazione viene effettuata aggiungendo al minuendo il complemento a due del sottraendo. Un sottrattore binario è, perciò, simile ad un sommatore al quale uno dei due numeri giunge

complementato (figura sottostante) e l'eventuale riporto, ottenuto dalla somma dei bit significativi, deve essere aggiunto alla somma di bit meno significativi. Ricordiamo che questo riporto si ha solo nel caso in cui il minuendo è maggiore del sottraendo e mai nel caso opposto e che in quest'ultimo caso il risultato è il complemento della differenza. Per chiarire ora verranno mostrati due esempi:

facendo il complemento a uno del risultato 1100, si ottiene 0011 che in decimale vale 3.

Sotto viene mostrato lo schema a blocchi del DM 7483 "4-bit binary adders with fast carry".

Se all'ingresso T viene inviato un 1, si ottiene la somma poichè il circuito combinatorio non complementa il numero B e blocca il passaggio del riporto da S5 a S0. Si ottiene, invece, una sottrazione se viene inviato in T uno 0, poichè il circuito combinatorio effettua la complementazione ed abilita al passaggio del riporto la porta P.

UNITÀ ARITMETICO-LOGICA (ALU)

Con l'aumentare delle richieste di sistemi digitali con sempre maggiore potenza di elaborazione, cui fa riscontro una maggiore facilità di realizzazione d'integrati LSI, sono state introdotte le unità aritmetico-logiche.

Questi circuiti sono in grado di compiere le necessarie funzioni logiche ed aritmetiche richieste per eseguire le varie istruzioni di un elaboratore. Esse operano normalmente con parole di quattro bit in codice binario o BCD, ma possono essere connesse in parallelo per aumentare la lunghezza della parola.

Per esempio, la MC 14581 B della Motorola è una ALU a CMOS che può compiere 16 operazioni aritmetiche su due parole di 4 bit, oppure 16 funzioni logiche. La scelta fra i due tipi di operazione è determinata dal livello presente all'ingresso "Mode control", mentre la funzione desiderata è selezionata applicando l'appropriata parola binaria agli ingressi di selezione S0, S1, S2, S3.

Sotto è mostrata la configurazione dei pin e le varie funzioni svolte da questa unità che è capace di pilotare due carichi TTL low-power, oppure uno TTL Schottky low-power, oppure due HTL.