



di GIORGIO OBER (GIOBE2000)

l'angolo di Mr A.KEER



(parte ottava)

PROGETTARE con le PORTE LOGICHE

Decoder Decimali, Encoder

Prosegue in questo numero l'analisi dei principali dispositivi combinatori; è la volta dei Decoder Decimali e degli Encoder, di cui scopriremo i segreti e tutti i componenti resi disponibili sul mercato.

Nella precedente puntata abbiamo analizzato i **Decoder Binari**, macchine combinatorie veramente importanti, non solo per la loro funzione come **Demultiplexer** ma anche per la frequente presenza nell'ambito del progetto digitale dedicato ai microprocessori; riprendiamo ora questi concetti con la descrizione per i **Decoder Decimali**, completando infine la trattazione con la loro struttura complementare: quella degli **Encoder**.

DECODER DECIMALI (TTL) : 74LS42, 7445, 74145, 7441, 74141

In generale i **Decoder** sono chiamati ad interpretare (= *decodificare*) le possibili *parole binarie* predisposte sulle loro linee d'ingresso e manifestano la loro azione provvedendo ad attivare *la sola* uscita (tra quelle previste) che corrisponde numericamente alla *parola* stessa; è ragionevole pensare che la loro quantità dipenda dal numero degli ingressi sui quali sono poste: con n ingressi ne saranno possibili 2^n , diverse tra loro.

In questo caso il loro insieme viene detto **codice binario puro a n bit** ed è esattamente quello che si troveranno ad interpretare i **Decoder Binari**, trattati ampiamente la puntata scorsa; la teoria booleana definisce numerosi *codici binari*, creati per garantire la sicurezza e la gestione dell'informazione; per la gran parte di essi sono previste solo 10 *parole*, un sottoinsieme delle 16 assicurate dal nativo *codice binario puro a 4 bit*.

Tra essi il **codice BCD** (**B**inary **C**oded **D**ecimal) assolve al compito di *codificare in binario* ciascuno dei 10 *simboli del sistema di numerazione decimale* ed è pensato per fornire uno strumento in grado di interpretare in modo umano (decimale) i dati (binari) prodotti dalle macchine digitali.

Molti dispositivi sono chiamati ad interpretare il *codice BCD* e tra essi proprio i **Decoder Decimali**; la serie **TTL** prevede alcuni integrati, descritti dai *datasheet* come **BCD to Decimal Decoder** o anche **4 Line to 10 Decimal Decoder**; tra essi il **74LS42**, del quale la *Figura 1* mostra il *pin-out*.

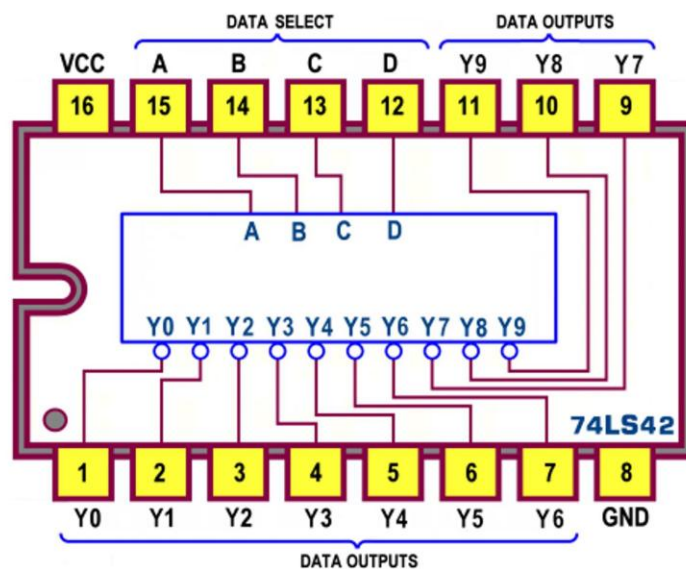


Figura 1 - 4-Line BCD To 10-Line Decimal Decoder 74LS42: Pin-out

Lo *schema funzionale* (o *schema logico*, *Logic Diagram* o *schematic circuit*) è decisamente interessante ma piuttosto complesso; esso consente un'analisi dettagliata del funzionamento, aiutandoci a capire l'influenza di ciascun segnale, seguendone il percorso. La *Figura 2* ne mostra i dettagli.

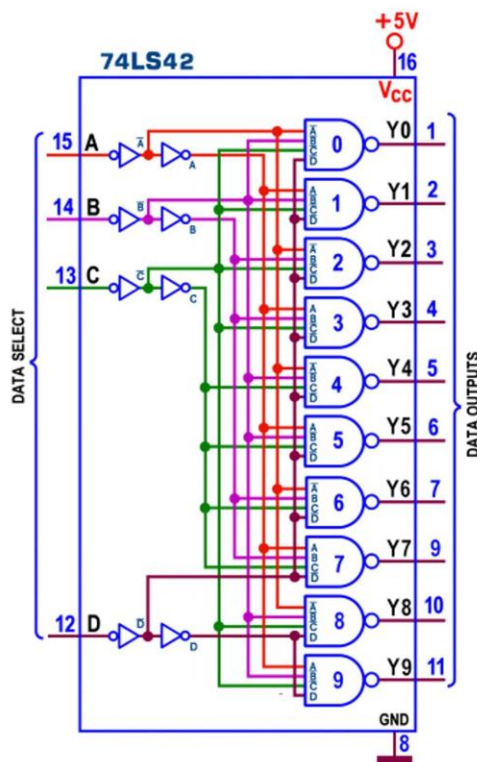


Figura 2 - 4-Line BCD To 10-Line Decimal Decoder 74LS42: Schema funzionale

Osservando la sua struttura risulta evidente che esso non ha tante uscite quante sono le possibili combinazioni binarie degli ingressi: disponendo di 4 ingressi di selezione (*Select*, A, B, C e D) secondo questa logica dovrebbe avere $2^4=16$ uscite; in realtà già sappiamo che questo compito è assicurato dal **Decoder Binario 74LS154**.

Il funzionamento di quest'ultimo è comunque molto simile a quello del **74LS42** e consiste sempre nell'attivazione di *una sola* linea d'uscita alla volta, con una importante differenza: il decoder decimale non dispone del segnale di controllo (*enable*).

Come si vede ciascuna delle uscite è supportata da una NAND per cui è facile capire che sarà attiva (*bassa*) solo quella la cui NAND *dispone di un 1* su tutti e 4 i suoi ingressi.

In pratica, in virtù della presenza di una rete logica (detta appunto di *decodifica*) appositamente studiata per questo scopo, sarà attiva solo quella identificata dal numero decimale 2^n corrispondente al codice BCD fornito sugli n ingressi di selezione.

Torniamo ora un momento sul fatto che questo dispositivo non dispone del segnale di *enable*; in realtà se tutti gli ingressi risultassero scollegati (o fluttuanti) *sentirebbero* un 1 logico e, poichè la sequenza binaria 1111 non appartiene al codice BCD, nessuna delle uscite risulterebbe attiva (*bassa*), il che equivale a disabilitare il decoder decimale!

L'opportunità appena rimarcata è molto utile quando dispositivi come questo sono controllati da un processore: supponendo di collegare agli ingressi di selezione le 4 linee meno significative del suo *bus dati*, per rendere disattivo il **74LS42** è sufficiente spedirgli una qualunque delle 6 parole non previste dal codice BCD, cioè quelle da 1010 a 1111.

In presenza delle altre 10 combinazioni *tutte le uscite* sono a 1 *meno una*, evidenziando la capacità del **74LS42** di saper riconoscere uno tra i possibili codici BCD forniti in ingresso.

74LS42															
INPUT					n	OUTPUT									
SELECT				Y0		Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	
D	C	B	A												
L	L	L	L	0	L	H	H	H	H	H	H	H	H	H	H
L	L	L	H	1	H	L	H	H	H	H	H	H	H	H	H
L	L	H	L	2	H	H	L	H	H	H	H	H	H	H	H
L	L	H	H	3	H	H	H	L	H	H	H	H	H	H	H
L	H	L	L	4	H	H	H	H	L	H	H	H	H	H	H
L	H	L	H	5	H	H	H	H	H	L	H	H	H	H	H
L	H	H	L	6	H	H	H	H	H	H	L	H	H	H	H
L	H	H	H	7	H	H	H	H	H	H	H	L	H	H	H
H	L	L	L	8	H	H	H	H	H	H	H	H	L	H	H
H	L	L	H	9	H	H	H	H	H	H	H	H	H	L	H
H	L	H	L	10	H	H	H	H	H	H	H	H	H	H	H
H	L	H	H	11	H	H	H	H	H	H	H	H	H	H	H
H	H	L	L	12	H	H	H	H	H	H	H	H	H	H	H
H	H	L	H	13	H	H	H	H	H	H	H	H	H	H	H
H	H	H	L	14	H	H	H	H	H	H	H	H	H	H	H
H	H	H	H	15	H	H	H	H	H	H	H	H	H	H	H

H livello logico **alto**
L livello logico **basso**

Figura 3 - 4-Line BCD To 10-Line Decimal Decoder 74LS42: Tabella di verità

La *Figura 3* presenta la *Tabella di Verità (Function Table)*, chiamata a sintetizzare le considerazioni appena fatte; risulta evidente l'assenza di uscite attive (*basse*) in corrispondenza delle codifiche non valide (... le ultime 6): poichè ciascuna di esse è di fatto *spenta*, in questa situazione si sul dire che siamo in presenza del *blanking* delle uscite.

Lo *schema pratico* (Figura 4) offre il simbolo del **74LS42**, da utilizzare direttamente nel contesto di un progetto, utile per evidenziare il nome dei *segnali logici* che entrano e di quelli che escono, e il *numero* dei piedini ad essi riservati.

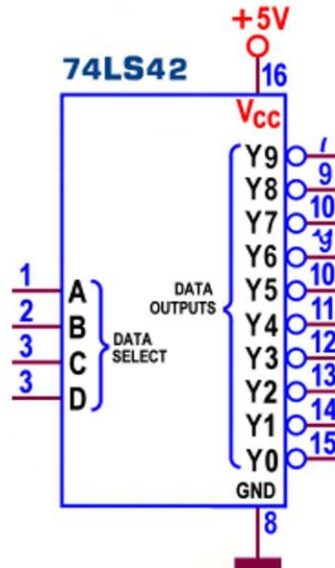


Figura 4 - 4-Line BCD To 10-Line Decimal Decoder 74LS42: Schema pratico

Il *simbolo logico* predisposto dallo *standard IEEE* è mostrato in Figura 5: come per gli altri *decoder* la notazione identifica i 4 ingressi di selezione (A, B, C e D) con il numero che indica il rispettivo *peso*, cioè l'esponente della potenza di 2 che stabilisce quale delle possibili uscite sarà abilitata.

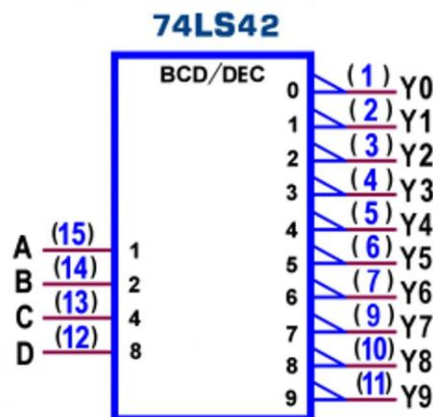


Figura 5 - 4-Line BCD To 10-Line Decimal Decoder 74LS42: Simbolo logico ANSI/IEEE Std. 91-1984

Può essere utile sottolineare che qualunque componente di questa categoria si presta, all'occorrenza, ad essere utilizzato come un *decoder binario "da 3 a 8"*.

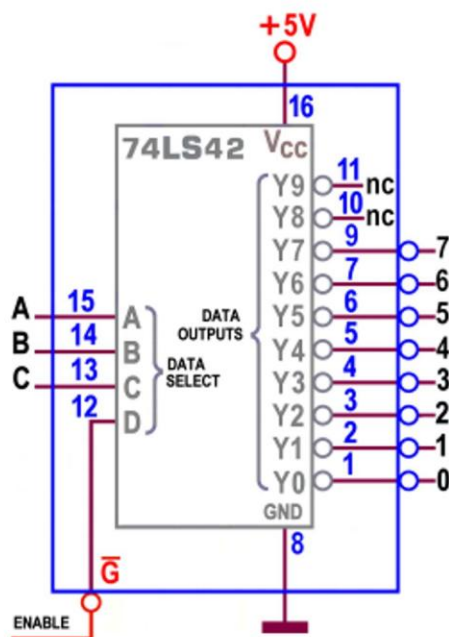


Figura 6 - 4-Line BCD To 10-Line Decimal Decoder 74LS42: uso come 3-Line To 8-Line Binary Decoder

Osservando la *Figura 6* è facile constatare che, scartando le 2 uscite più significative, per controllare le rimanenti 8 bastano le 3 linee di selezione A, B e C, purchè la quarta linea D sia posta a 0 logico; se essa è lasciata fluttuante o scollegata tutte le uscite sono forzate *alte*, il che equivale ad affidare a D una chiara azione da *enable*.

La corrente massima assorbita dalle uscite del **74LS42** è quella tipica **TTL LS** ($I_{OL}=8\text{ mA}$) mentre la versione **TTL Standard** (**7442**) assicura valori doppi ($I_{OL}=16\text{ mA}$); la serie **TTL** prevede altri integrati per questa categoria, ad esso funzionalmente identici e tutti con esso *pin-out compatibili*: il **7445** e il **74LS145**, entrambi a *collettore aperto*, sono definiti **decoder/driver** per l'eccellente capacità di pilotaggio delle loro uscite (fino a **80 mA**), garantite dalla presenza di transistor ad alta prestazione ideali per pilotare direttamente *lampade ad incandescenza* o *relè*.

Il **74LS537** è una versione sofisticata, dotata di 20 piedini; il componente è abilitato dalla presenza contemporanea di 2 segnali (uno *attivo alto* e l'altro *attivo basso*, concorrenti in una **AND**) che consentono il passaggio del codice BCD applicato ai 4 ingressi; le 10 uscite, di tipo *three-state*, possono essere *attive alte* o *basse*, in funzione del valore di un apposito segnale di controllo.

Il *ritardo di propagazione* massimo (*Propagation Delay Time*, con carico di **400ohm/15pF**) è mediamente dai **30 ns** ai **50 ns** per tutte le transizioni e per tutti i modelli, mentre la potenza dissipata massima (con uscite aperte e ingressi a **0**) è di **65 mW** per i **TTL LS** e va da **280 mW** a **350 mW** per quelli **TTL Standard**.

Un particolare cenno meritano il **7441** e il **74141** specializzati nella gestione diretta delle fiammeggianti *nixie*, improbabili tubi di vetro, piccole valvole con catodi a forma di numero, posti uno dietro l'altro; l'alone luminoso generato dalla scarica sul gas in essi contenuto rimbalza avanti e indietro, illuminando di rosso la cifra coinvolta, in una danza mai vista prima..

C'è ancora una discreta schiera di appassionati di elettronica che si cimentano in progetti "dal sapore antico": sebbene sia i componenti che questi magici visualizzatori siano obsoleti da

molti anni non per questo sono introvabili, frequentando i mercatini, consultando qualche negozio specializzato o più semplicemente cercando su **eBay**..

La Figura 7 mostra il pin-out del **7441** e Figura 8 mostra il schema pratico.

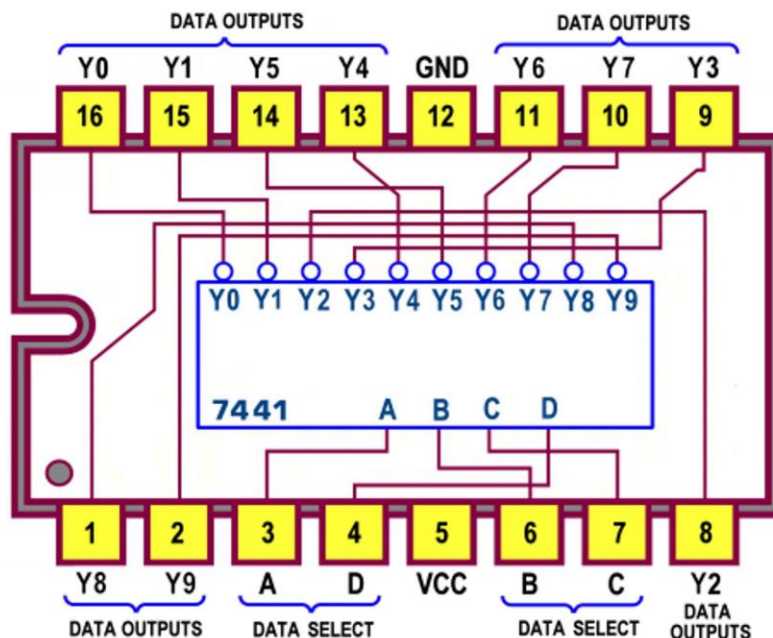


Figura 7 - BCD To Decimal Decoder Nixie Driver 7441: Pin-out

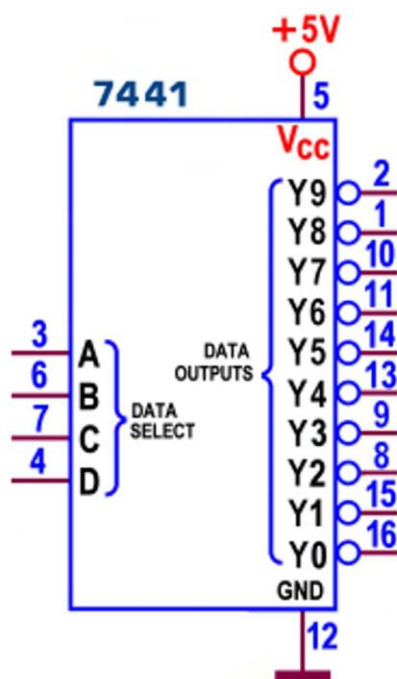


Figura 8 - BCD To Decimal Decoder Nixie Driver 7441: Schema pratico

Salta subito agli occhi il modo incongruente di associare le funzioni interne ai piedini dell'integrato, molto approssimativa: per quello che concerne le uscite e gli ingressi non c'è alcuna consequenzialità mentre i 2 pin destinati all'alimentazione (**Vcc** sul pin 5 e **GND** sul pin 12) sono collocati in *posti inconsueti* rispetto agli integrati TTL moderni. Entrambe le considerazioni confermano una certa vetustà del componente.

La *Figura 9* mostra il *simbolo logico* predisposto dallo *standard IEEE*, osservando il quale possiamo notare alcune novità: il simbolino a firma di triangolo, sotto la definizione di categoria BCD/DEC, per indicare la sua funzione di *driver*; e il piccolo rombo su ciascuna uscita, per indicare la sua natura *open collector*.

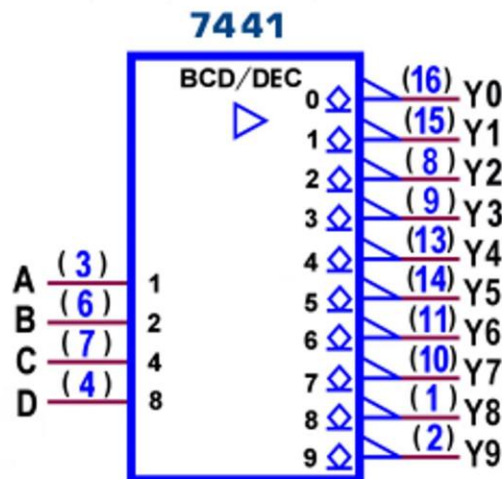


Figura 9 - BCD To Decimal Decoder Nixie Driver 7441: Simbolo logico ANSI/IEEE Std. 91-1984

Tutte le *Figure* sono valide anche per il **74141**, più affidabile del suo predecessore perchè (tra l'altro) supporta correttamente la funzione di *blanking*, cioè tiene *spente* (a *livello alto*) le uscite se sui suoi ingressi sono fornite le 6 parole non previste dal codice BCD, cioè quelle da 1010 a 1111, in perfetto accordo con la Tabella di Verità di *Figura 3*; sfortunatamente ciò non è stato previsto per il **7441** per cui, non appena viene interessato da questi codici, esso attiva lo stesso almeno una delle sue uscite, secondo una logica legata alla sua data di produzione, identificata dalla presenza di una lettera di revisione, A o B, dopo la sigla.

In aggiunta il **74141** dispone sulle uscite di diodi zener di protezione da 55V: non va dimenticato che questi componenti sono chiamati a gestire tensioni del tutto inconsuete per il normale governo TTL, fino ai 170V, da porre sull'anodo dalle nixie per il loro funzionamento.

Prima di terminare è curioso sottolineare la presenza di altri 2 componenti da ritenere appartenenti al gruppo, per i quali rimangono validi il *pinout*, lo *schema pratico* e il *simbolo ansi/ieee* del **74LS42**; sono entrambi a tutti gli effetti **4-Line To 10-Line Decimal Decoder** ma differiscono da quelli trattati finora per il fatto di attivare (*basse*) le 10 uscite a partire da un codice diverso dal **BCD**, sia pure sempre costituito da 10 parole a 4 bit. Il **74LS43** interpreta un codice **Eccesso a 3** e il **74LS44** un codice **Eccesso a 3 Gray** (una variante ridotta del codice Gray vero e proprio).

La *Figura 10* mette a confronto le sequenze d'ingresso nei 3 casi trattati; per tutti vale l'opportunità di forzare il *blanking* delle uscite nel caso siano fornite le 6 codifiche eccedenti le prime 10 (uniche previste dal codice che li governa).

n	74LS42	74LS43	74LS44
	BCD	ECCESSO A 3	ECCESSO A 3 GRAY
	SELECT	SELECT	SELECT
	D C B A	D C B A	D C B A
0	L L L L	L L H H	L L H L
1	L L L H	L H L L	L H H L
2	L L H L	L H L H	L H H H
3	L L H H	L H H L	L H L H
4	L H L L	L H H H	L H L L
5	L H L H	H L L L	H H L L
6	L H H L	H L L H	H H L H
7	L H H H	H L H L	H H H H
8	H L L L	H L H H	H H H L
9	H L L H	H H L L	H L H L
10	H L H L	H H L H	H L H H
11	H L H H	H H H L	H L L H
12	H H L L	H H H H	H L L L
13	H H L H	L L L L	L L L L
14	H H H L	L L L H	L L L H
15	H H H H	L L H L	L L H H

H livello logico **alto**
L livello logico **basso**

Figura 10 - Decimal Decoder 74LS42, 74LS43, 74LS44: Ingressi di selezione

Il codice **eccesso 3** deriva il suo nome dal fatto di coinvolgere la stessa sequenza di parole del BCD a partire da quella associata al numero 3; esso è nato per facilitare il trattamento dei dati in progetti (come le calcolatrici) che acquisiscono e mostrano informazioni nel sistema di numerazione decimale dovendo (ovviamente) elaborarle in quello binario; il passaggio (conversione) da un sistema all'altro richiede il coinvolgimento di *tecniche di complemento* delle cifre che possono essere evitate codificandole con questo codice *autocomplementante*; con riferimento alla linea mediana della tabella si può notare infatti che dalle prime 5 parole si ottengono le altre 5, scambiando gli 1 con gli 0: la parola binaria (0011) associata allo 0 è il *complemento a 1* di quella (1100) associata al 9, e così l'1 (0100) con l'8 (1011), ecc...

Il codice **Gray** è stato creato per tentare di risolvere un problema che si è dimostrato critico nel controllo numerico dell'informazione binaria: il contenuto di un registro è soggetto a continue variazioni che, nel caso più impegnativo, può cambiare *tutti* i suoi bit interni; per esempio, nel passaggio da 7 (0111) a 8 (1000), sebbene il segnale di sincronismo (*clock*) sia portato nello stesso istante a tutte le celle del registro, è impossibile che le 4 cifre cambino contemporaneamente, perchè oggettivamente un circuito può essere più veloce dell'altro, sebbene per un tempo molto piccolo (dell'ordine dei ns). Per risolvere il problema questo codice fa in modo che alle cifre contigue sia affidata una parola binaria che differisce per un solo bit, evitando che si possa presentare la situazione critica appena descritta. A contorno possiamo notare che, nella versione ridotta **Eccesso a 3 Gray** utilizzata dal componente, i 3 bit meno significativi delle sue parole sono *riflessi* rispetto alla mediana della tabella, mentre il bit più pesante è sempre uguale (o 0 o 1) nelle 2 metà.

DECODER DECIMALI (CMOS) : 4028

Anche la serie **CMOS 4000** prevede un **BCD to Decimal Decoder**: si tratta del **4028**, funzionalmente identico a quelli TTL; la *Figura 11* mostra il suo *pin-out* e *Figura 12* il suo *schema pratico*.

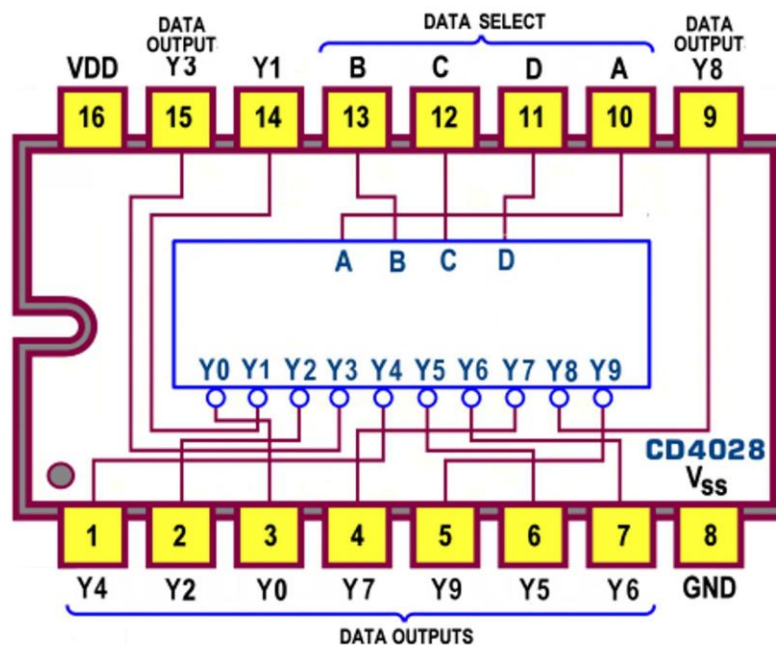


Figura 11 - 4-Line BCD To 10-Line Decimal Decoder 4028: Pin-out

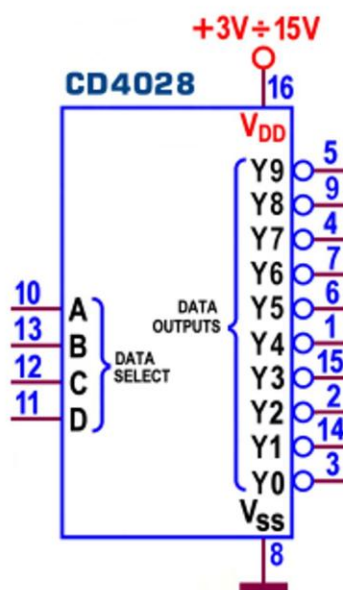


Figura 12 - 4-Line BCD To 10-Line Decimal Decoder 4028: Schema pratico

ENCODER (TTL) : 74LS147, 74LS148

Come suggerisce il loro nome, gli **Encoder** sono chiamati ad esercitare il compito inverso di quello garantito dai **Decoder**: la presenza di un livello logico attivo su una qualunque delle sue *linee d'ingresso* è rilevata e codificata sulle sue *linee d'uscita* con una *parola binaria* che esprime il suo numero d'ordine; con 8 ingressi (numerati da 0 a 7) saranno dunque necessarie 3 uscite sulle quali, all'occorrenza, saranno fissate le rispettive sequenze binarie (per esempio da 000 a 111).

Il meccanismo è decisamente interessante ma poco pratico: vale solo se è attivo un unico ingresso alla volta; con 2 o più ingressi attivi contemporaneamente la codifica prodotta in uscita dalla macchina combinatoria risulta inutilizzabile e del tutto inattendibile.

Per questa ragione sono state introdotte delle varianti di progetto che hanno reso impossibile questa situazione, assicurando una *logica di priorità* in grado di garantire la codifica *solo* all'ingresso di peso maggiore: tutti gli eventuali ingressi attivi con priorità inferiore saranno ignorati.

La serie **TTL LS** prevede 2 integrati (entrambi prodotti anche nella serie **Standard**), descritti dai *datasheet* come **Priority Encoder**: il **74LS147**, con 9 linee in ingresso e 4 uscite, e il **74LS148**, con 8 linee in ingresso e 3 uscite; solo quest'ultimo, più sofisticato degli altri, è ancora in produzione; le Figure 13 e 14 mostrano i rispettivi *pin-out*.

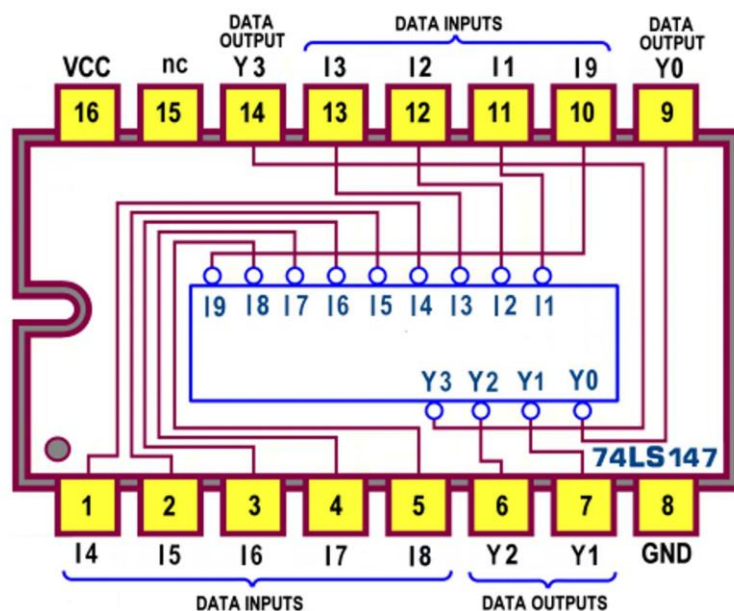


Figura 13 - 10-to-4 Line Priority Encoder 74LS147: Pin-out

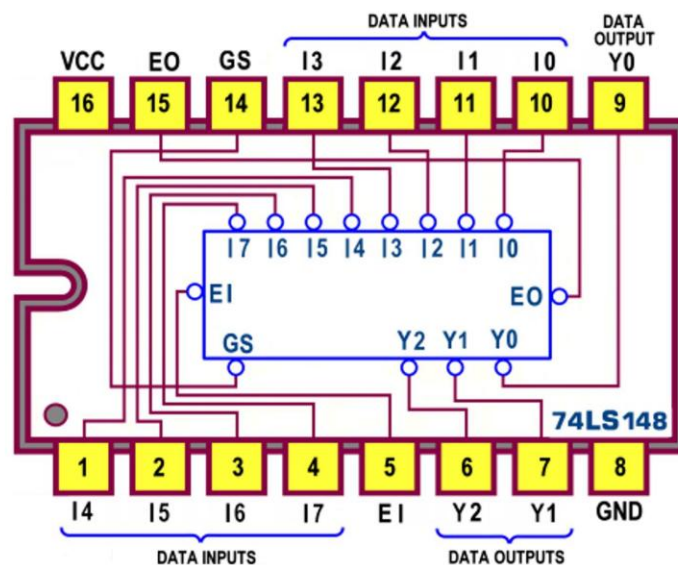


Figura 14 - 8-to-3 Line Priority Encoder 74LS148: Pin-out

Lo *schema funzionale* (o *schema logico*) è molto interessante ma, al solito, piuttosto complesso per cui non si ritiene opportuno aggiungerlo a questa trattazione, lasciando al lettore il compito di scoprirlo sui *datasheet*; di fatto esso mostra su ognuna delle 3 o 4 uscite una struttura di tipo **AND-OR-NOT** in grado di raccogliere l'influenza degli 8 o 9 ingressi al fine di fissare il livello logico necessario per codificare in binario (a 3 o a 4 bit) il numero d'ordine decimale dell'ingresso con maggiore priorità.

L'analisi dello *schema pratico* ci dà l'occasione di riprendere la descrizione del funzionamento dei 2 dispositivi; la Figura 15 mostra il simbolo (utile nel contesto dei nostri progetti) del **74LS147**.

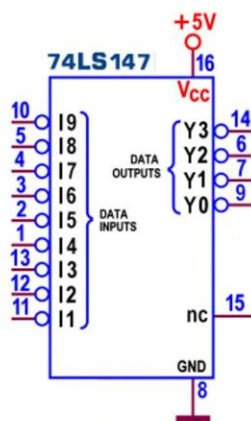


Figura 15 - 10-to-4 Line Priority Encoder 74LS147: Schema pratico

Nonostante la presenza di soli 9 ingressi *attivi bassi*, essi sono sufficienti per garantire sulle 4 uscite (esse pure *attive basse*) tutte e 10 le parole del codice **BCD**; la decima linea non è infatti necessaria perché la codifica 0000 relativa al numero **0** (associabile all'ingresso **I0** non previsto) è implicitamente forzata sulle uscite (di fatto con valore 1111, per l'inversione logica da esse imposta) quando nessuna delle 9 linee d'ingresso risulta attiva, cioè quando sono tutte *alte*.

Anche per questo è definito **10-to-4 Line Priority Encoder** dai *datasheet*; la priorità massima è affidata all'ingresso **19**: in ogni caso, quando 2 o più ingressi sono attivi contemporaneamente, il codice **BCD** (sottoposto a *complemento a 1*) presente in uscita sarà quello assegnato all'ingresso con la priorità più alta.

L'analisi della *Tabella di Verità* (*Function Table*, *Figura 16*) conferma il modo di operare del componente.

74LS147													
INPUT									OUTPUT				
LINEE INGRESSO									Y3	Y2	Y1	Y0	
1	2	3	4	5	6	7	8	9					
H	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	L
X	X	X	X	X	X	X	L	H	L	H	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H	H
X	X	X	X	L	H	H	H	H	H	L	H	L	L
X	X	X	L	H	H	H	H	H	H	L	H	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H	H
L	H	H	H	H	H	H	H	H	H	H	H	L	L

H livello logico **alto**
L livello logico **basso**
X livello logico **irrelevante**

Figura 16 - 10-to-4 Line Priority Encoder 74LS147: Tabella di verità

La *Figura 17* mostra il simbolo del **74LS148**: definito **8-to-3 Line Priority Encoder**.

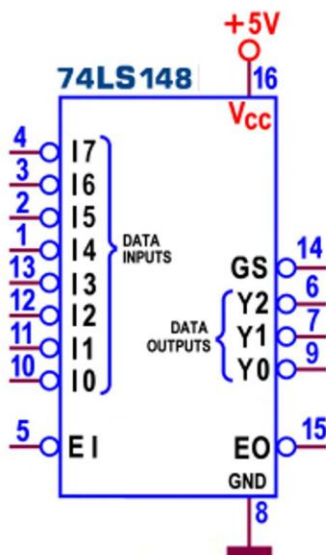


Figura 17 - 8-to-3 Line Priority Encoder 74LS148: Schema pratico

Oltre alle 8 linee d'ingresso e alle 3 linee d'uscita (tutte *attive basse*) si notano 3 linee di controllo; se forzato a **0** logico il segnale *Enable Input* (**EI**, *attivo basso*) consente il funzionamento (*abilita*) dell'encoder e, in caso contrario, forza *alte* tutte e 5 le uscite; l'uscita *Enable output* (**EO**, *attiva bassa*) va a **0** (supposto il dispositivo abilitato, cioè con **EI**=0) solo se nessuna delle linee d'ingresso risulta attiva; l'uscita *Group signal* (**GS**, *attiva bassa*) segnala con uno **0** la presenza di almeno un ingresso attivo, ovviamente sempre se l'encoder è abilitato.

Risulta chiaro che la codifica proposta sulle uscite è del tipo *binario puro a 3 bit* e che la priorità massima è affidata all'ingresso **I7**; ma la presenza delle 3 linee di controllo rende molto versatile questo componente, favorendo la sua espandibilità; la *Figura 18* mostra lo schema che utilizza 2 **74LS148** in cascata per governare 16 ingressi con priorità.

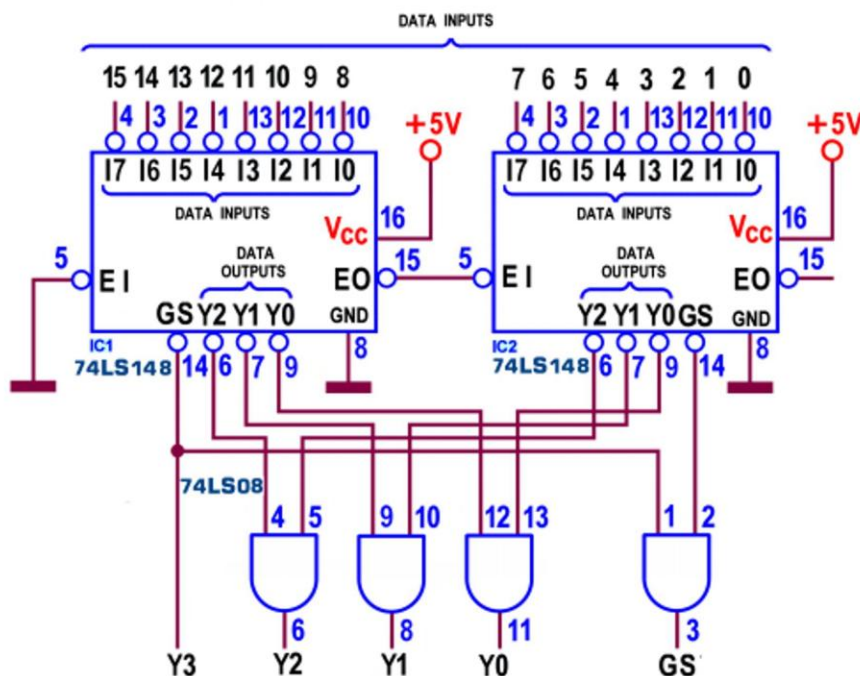


Figura 18 - 8-to-3 Line Priority Encoder 74LS148: Adattamento Encoder da 8 a 16 bit

In breve: l'abilitazione forzata **EI=0** dell'encoder IC1 che gestisce gli *ingressi più significativi* consente l'abilitazione dell'altro, IC2, solo se nessuno di essi risulta attivo; in questa situazione la sua **EO** vale **0** (e abilita il secondo encoder), le sue 3 uscite sono a 1 logico (e tengono aperte le rispettive **AND** autorizzando il passaggio del codice prodotto dal secondo, in funzione del rispettivo ingresso attivo, se c'è) e la sua **GS** vale **1** e costituisce il bit più significativo della quaterna di bit finale, del tipo **1xxx**.

Se invece sul primo encoder IC1 è attivo almeno un ingresso la sua **EO** vale **1**, disabilitando il secondo; sono le 3 uscite di quest'ultimo ad essere ora a **1** logico, tenendo a loro volta aperte le rispettive **AND** e autorizzando ora il passaggio del codice prodotto dal primo, la **GS** del quale vale **0**, coerentemente con il fatto che ora la quaterna finale di bit sarà di tipo **0xxx**. Non dimentichiamo che le 4 uscite di dato sono *attive basse* per cui, se almeno uno dei 16 ingressi è attivo, il codice presente su di esse andrà da **1111 (=I0)** a **0000 (=I15)**; in assenza di ingressi attivi tutte le 5 uscite finali sono a **1**.

Mai come in questo caso la *Tabella di Verità* (*Function Table*, *Figura 19*) permette di capire meglio le considerazioni appena fatte; in particolare è utile valutare con pazienza l'azione dell'abilitazione **EI** e l'effetto degli 8 ingressi sulle uscite di codice (sempre esprimanti il *complemento a 1* del numero binario associato all'ingresso con maggiore priorità), su **GS** e su **EO**.

74LS148														
INPUT										OUTPUT				
LINEE INGRESSO									EI	Y2	Y1	Y0	GS	EO
0	1	2	3	4	5	6	7							
X	X	X	X	X	X	X	X	X	H	H	H	H	H	H
H	H	H	H	H	H	H	H	H	L	H	H	H	H	L
X	X	X	X	X	X	X	X	L	L	L	L	L	L	H
X	X	X	X	X	X	L	H	H	L	L	L	H	L	H
X	X	X	X	X	L	H	H	H	L	L	H	L	L	H
X	X	X	X	L	H	H	H	H	L	L	H	H	L	H
X	X	X	L	H	H	H	H	H	L	H	L	L	L	H
X	X	L	H	H	H	H	H	H	L	H	L	H	L	H
X	L	H	H	H	H	H	H	H	L	H	H	L	L	H
L	H	H	H	H	H	H	H	H	L	H	H	H	L	H

H livello logico **alto**
L livello logico **basso**
X livello logico **irrelevante**

Figura 19 - 8-to-3 Line Priority Encoder 74LS148: Tabella di verità

Il simbolo logico standard IEEE dei 2 componenti è mostrato rispettivamente in Figura 20 e in Figura 21.

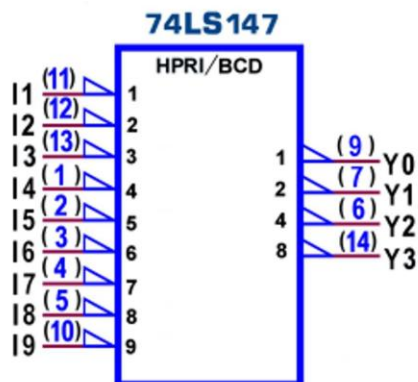


Figura 20 - 10-to-4 Line Priority Encoder 74LS147: Simbolo logico ANSI/IEEE Std. 91-1984

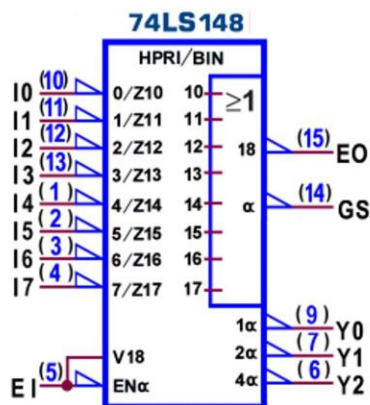


Figura 21 - 8-to-3 Line Priority Encoder 74LS148: Simbolo logico ANSI/IEEE Std. 91-1984

La specificità di entrambi viene rimarcata dalla sigla **HPRI** (**H**ighest **PRI**ority) e dalla segnalazione relativa alla codifica prodotta (**BCD** nel primo caso e **BIN**aria nel secondo); francamente assillante la pignola dotazione di dettagli proposta dal secondo per tener conto di ogni particolare del suo funzionamento.

Le caratteristiche elettriche dei 2 encoder **TTL LS** sono quelle della *famiglia*: tra esse $I_{OL}=8$ **mA** (corrente massima *assorbita* in uscita a livello **0**) e $V_{OH}=3,5$ **V** (tensione *tipica* in uscita a livello **1**); la potenza dissipata massima (con uscite aperte e ingressi a **0**) è di **100 mW** e il *ritardo di propagazione* massimo (*Propagation Delay Time*, con carico di **2kohm/15pF**) va da **17 ns** a **55 ns** per numerose transizioni fornite dai manuali; nella versione **TTL standard**, oltre a disporre di correnti massime più elevate ($I_{OL}=16$ **mA**, *assorbita* in uscita) dissipano al massimo **300 mW** ed oppongono un *ritardo* massimo di **19 ns** con carico di **400ohm/15pF**.

La serie **TTL LS** prevede altri integrati per la categoria dei encoder, tutti *pin-out compatibili* con il **74LS148**: nel **74LS348** le 3 uscite di codice sono di tipo *three-state*, forzate in alta impedenza se il chip non è abilitato ($EI=1$) oppure se (con $EI=0$) nessuna delle linee d'ingresso risulta attiva; il **74LS748** e il **74LS848** sono invece varianti proprietarie della Motorola immesse sul mercato per porre rimedio ad un problema legato alla natura combinatoria di questi componenti per il quale, sul fronte di discesa dell'abilitazione **EI** e in assenza di ingressi attivi, si può manifestare un breve indesiderato impulso (*basso*) spurio (*glitch*) su **GS** o la momentanea abilitazione delle 3 uscite di codice, nonostante in queste condizioni tutte e 4 queste uscite dovrebbero rimanere a 1.

ENCODER (CMOS) : 4532

La serie **CMOS 4000** prevede il **4532**, funzionalmente identico e *pinout* compatibile con il **74LS148**; la Figura 22 ne ripropone lo schema e la Figura 23 riassume le sue caratteristiche.

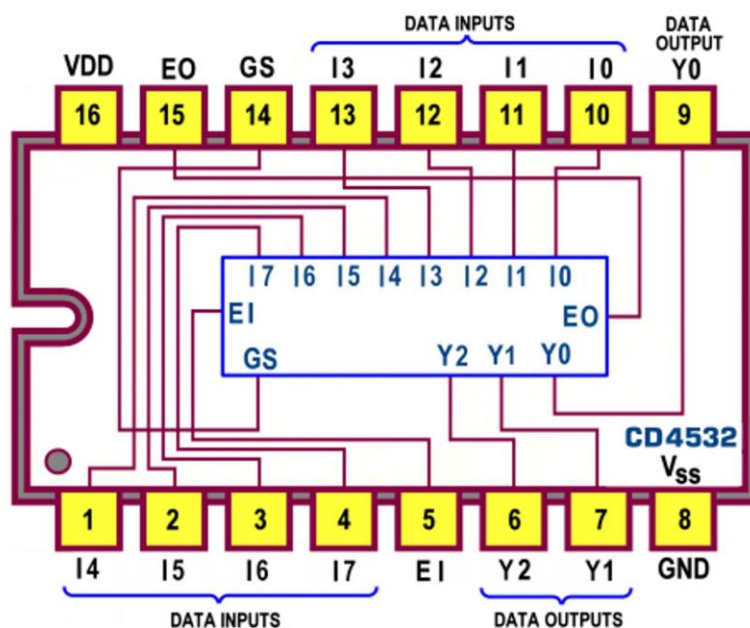


Figura 22 - 8-to-3 Line Priority Encoder 4532: Pin-out

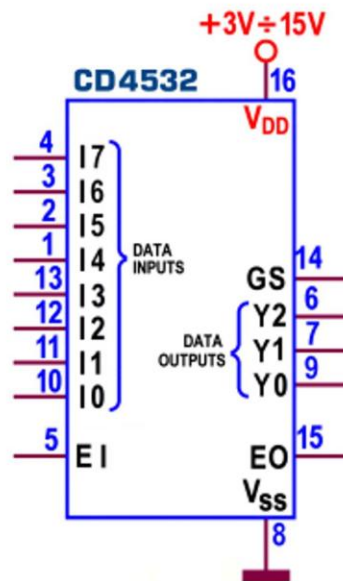


Figura 23 - 8-to-3 Line Priority Encoder 4532: Schema pratico

Si può notare che tutti i terminali sono ora *attivi alti*, sia gli 8 ingressi che le 3 uscite di codice, ma anche le 3 linee di controllo: la *Enable Input* deve ora essere forzata a **1** logico per consentire il funzionamento dell'encoder e, in caso contrario, forza *basse* tutte e 5 le uscite; l'uscita *Enable output* va a **1** (supposto il dispositivo abilitato, cioè con **EI**=1) solo se nessuna delle linee d'ingresso risulta attiva; l'uscita *Group signal* segnala con un **1** la presenza di almeno un ingresso attivo, ovviamente sempre se l'encoder è abilitato.