di GIORGIO OBER (GIOBE2000)

l'angolo di Mr A.KEER

(parte nona)

PROGETTARE con le PORTE LOGICHE

Visualizzatori "a 7 segmenti"

Dopo la ricca rassegna di Decoder Binari e Decimali non può mancare l'analisi dei principali dispositivi combinatori pensati per interfacciare il più "estetico" dei componenti: il visualizzatore numerico.

Una delle più importanti esigenze del progetto digitale è quella di proporre le informazioni in modo diretto e immediatamente interpretabile; per questo da lungo tempo sono stati creati i visualizzatori (display) ad una o più cifre (digit = numero).

Fin dalla loro prima apparizione, non solo in ambiente tecnico, hanno appagato il gusto estetico come poche altre cose; cosa sarebbe della suspense di un thriller se l'inevitabile (e inesorabile) bomba non esibisse il suo contatore con estenuante cadenza?

L'idea fu della americana Burroughs Corporation che nei primi anni '50 del secolo scorso introdusse le Nixie (ne abbiamo fatto cenno la puntata precedente), piccole fiammeggianti valvoline con catodi a forma di numero, posti uno dietro l'altro e illuminati dalla scarica elettrica sul gas in esse contenuto.

Per una ventina d'anni non c'era strumento elettronico o orologio o dispositivo di controllo numerico che non ne facesse uso! Naturalmente anche oggi sono numerosissimi i dispositivi con un display, molto più pratico dei primi che, dovendo disporre per l'accensione delle cifre di una tensione continua di circa **170V** tra griglia/anodo ed catodi, erano anche pericolosi e delicati. Sebbene ne esistano di diversa natura tecnologica, i più pratici ed economici sono certamente quelli allo stato solido, sostanzialmente a Led.

Un singolo digit è solitamente composto da (almeno) 8 led chiamati ad illuminare 7 piccole guide ottiche (segmenti) disposte in modo da formare la cifra "8", leggermente inclinata verso destra; l'ottavo led è utilizzato per gestire il punto decimale (*Decimal Point*).

E' chiaro fin d'ora che, per ogni digit, è necessario disporre di un'interfaccia in grado di tradurre un'informazione *binaria* (di solito un codice fornito da una logica booleana o elaborato da un processore) in una *simbolica*, adatta alla sensibilità umana, accendendo la formazione dei 7 segmenti in modo da produrre una delle 10 cifre del sistema di numerazione decimale, ma anche (con particolari accorgimenti) quasi tutte le lettere e numerosi simboli utili.

Da molto tempo questo problema è stato risolto dagli studiosi di logica digitale con il progetto una macchina combinatoria in grado di accettare in ingresso un codice binario a 4 bit e rendere

disponibili in uscita sette linee logiche adatte a controllare lo stato (acceso o spento) di altrettanti led.

Questa macchina è dunque un **Decoder**, cioè un circuito logico in grado di interpretare un codice e tradurre (decodificare) ogni sua parola in una azione specifica; ma prima di approfondire la loro conoscenza è opportuno scoprire gli oggetti che sarà chiamato a governare.

Digit "a 7 segmenti": FND500, FND507

Data la loro indispensabile necessità il mercato offre una vasta gamma di modelli, diversi per dimensione e/o per colore dei led; di solito il progettista si limita a scegliere questa periferica optoelettronica in funzione della sua forma estetica o delle proprie necessità d'ingombro, non curandosi (come invece succede per altri componenti) di conoscere il suo produttore.

Per questa ragione le sigle dei digit proposti come esempio in *Figura 1* sono sostanzialmente irrilevanti.

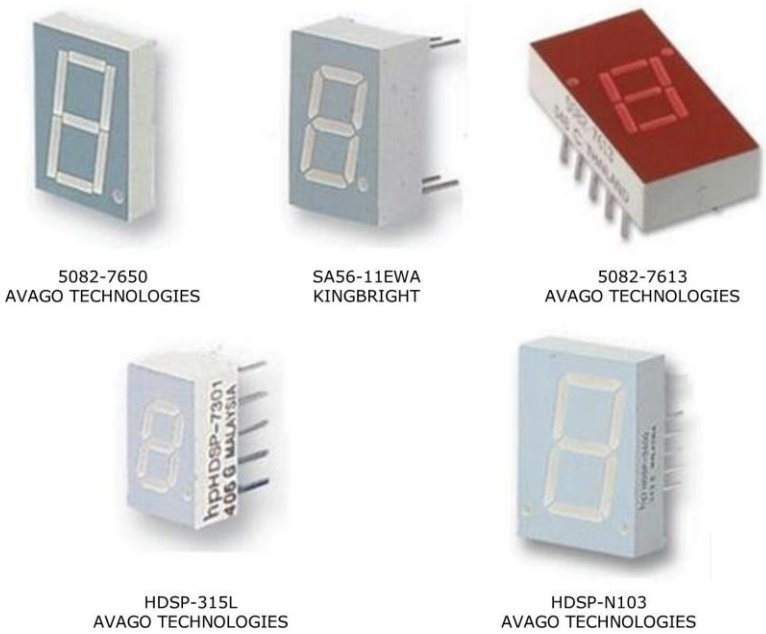


Figura 1 - Raccolta di digit commerciali

Esternamente si distinguono chiaramente sette piccoli segmenti sagomati, raggruppati per comporre un "8" ben evidente sulla plastica grigia o rossa, ma "dentro" è tutta una magia..

Lo strato più basso è, spesso, una piccola schedina di vetronite (della dimensione del componente, *Figura 2a*) sulla quale si notano (a fatica) 8 minuscoli puntini di un certo spessore, collegati tra loro da sottili piste che partono e arrivano ai piedini di metallo, posti su due lati opposti; naturalmente si tratta di un piccolo deposito di semiconduttore (di solito arseniuro o fosfuro di gallio). Sottoponendola, con attenzione, alla corretta differenza di potenziale, la piccola massa si accende, di solito con un'intensità luminosa decisamente deludente.

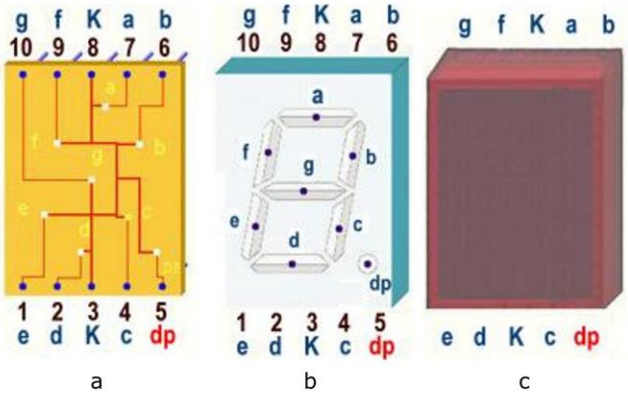


Figura 2 - Single Digit Numeric Display: Struttura

Subito sopra si appoggia un parallelepipedo di plastica opaca, di superficie uguale a quella della schedina e alto 3 o 4 mm (*Figura 2b*); in esso sono incisi dei piccoli solchi a sezione triangolare, con le pareti "a specchio" e un piccolo foro al centro, dal quale emerge il semiconduttore. Applicando di nuovo la corretta tensione, la piccola luce tende a riflettersi in tutta la dimensione del solco, senza ancora particolare visibilità.

L'ultimo elemento del componente (*Figura 2*c) è una scatolina di plastica trasparente rossa, chiamata a contenere le due parti precedenti; il suo compito però non si limita a questo: la superficie interna del suo fondo è coperta di minuscole piramidi a sezione quadrata, in pratica un *catarifrangente* come quelli delle luci posteriori dei nostri veicoli. Poiché la sua arte è quella di riflettere la luce nella stessa direzione da cui è scaturita, le infinite riflessioni sulla sottostante guida ottica riflettente si estendono in un attimo a tutta la dimensione del solco, integrando e filtrando la luce e facendole assumere la dignità di segmento.

La figura mette in evidenza anche un concetto fondamentale nell'utilizzo dei digit: la convenzione che assegna a ciascun segmento una ben precisa lettera, dalla "**a**" alla "**f**" in senso orario a partire dall'alto, con la "**g**" in posizione centrale.

Uno dei più diffusi digit sul mercato è il **FDN500** della **Fairchild**: esso rispetta la logica costruttiva appena descritta ma offre una soluzione tecnologica più pregiata: ora la schedina di vetronite è sostituita da un agglomerato di plastica nera (della stessa dimensione) nella quale è affogata una serie di robusti lamierini sagomati con la forma dei collegamenti interni, in parte terminati con i dieci piedini da esso emergenti, ovviamente ad angolo retto, rispetto ai collegamenti stessi.

Ciascuno degli 8 elementi fotoemittenti è ora fisicamente un *microled*, simile ad un mezzo disco di plastica trasparente, di 3 mm di diametro e spesso 1 millimetro, col chiaro compito di fare da lente al semiconduttore, posto esattamente al centro della sua base.

Il dettaglio fotografico (*Figura* 3) mostra l'angolo in basso a destra della resina nera: si distinguono quattro lamierini che (sul lato inferiore) continueranno verso l'esterno sottoforma di piedini (pin5 = decimal point, pin4 = segmento **c**, pin3 = catodo, pin2 = segmento **d**); il più grande è ovviamente quello associato al *catodo comune* e, da esso verso gli altri, si notano i tre microled relativi agli elementi del digit appena citati.

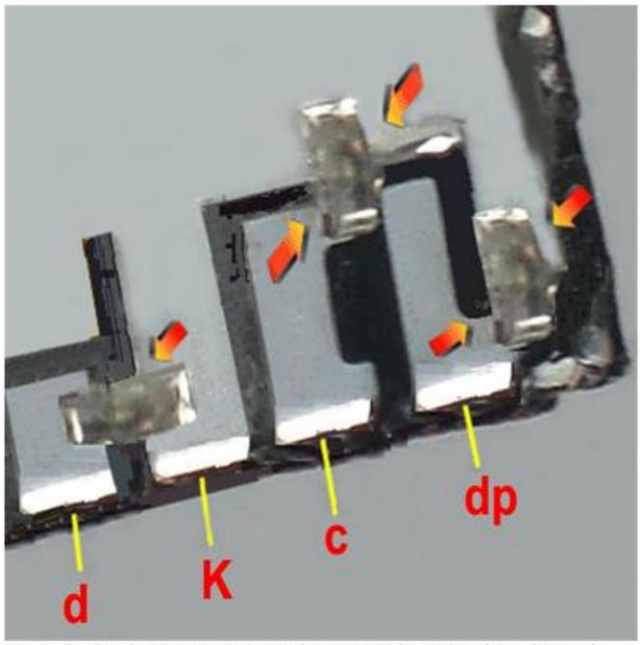
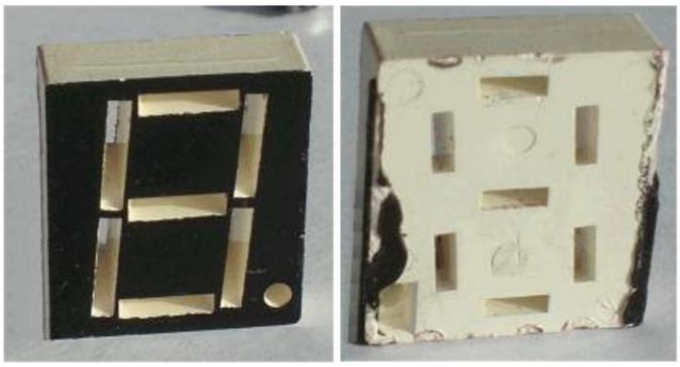


Figura 3 - Single Digit Numeric Display FND500/FND507: dettaglio Hardware

La guida ottica centrale, molto più spartana di quella descritta in precedenza; da un lato (*Figura* 4a) accoglie ora i microled, offrendo fessure 3x1 mm esattamente adatte ad ospitarli, e, dal lato opposto (*Figura* 4b) , porta le fessure alla dimensione finale di ciascun segmento, 6x1,5 mm.



a b

Figura 4 - Single Digit Numeric Display FND500/FND507: dettaglio Hardware

L'effetto lente garantito dalla struttura dei microled rende inutile la presenza del catarifrangente sulla superficie interna dell'involucro di plastica rossa, che risulta pertanto liscio e trasparente; esso mantiene comunque il compito di esaltare la componente rossa della luce emessa, garantendo in fine un eccellente risultato.

I digit a Led sono disponibili in due forme funzionali alternative, dette a ***catodo comune*** e ad ***anodo comune***; noi tutti sappiamo che i diodi a semiconduttore consentono il passaggio di una corrente diretta, se sottoposti ad una corretta differenza di potenziale; la caduta di tensione ai capi di un diodo LED dipende dalla sua corrente di lavoro ma anche dal suo colore; escludendo i Led Flash di ultima generazione (bianchi o blu, che assorbono fino a **40mA** con cadute tipiche di **3V**, di solito non utilizzati nei digit) nessuno dei rimanenti tipi potrà

accendersi con tensioni applicate minori di **1,6V** (la cosiddetta *tensione di soglia*, molto maggiore di quella di un diodo normale al silicio, pari a **0,6V**) mentre la tensione di lavoro, con una corrente tipica dai **10mA** ai **20mA**, è tipicamente di **1,7V** per il Rosso, **1,9V** per il Giallo, **2,0V** per il Verde e per l'Arancio.

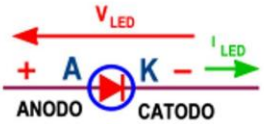


Figura 5 - Diodo Led: schema

Di certo la *corrente convenzionale* scorre dal polo positivo verso quello negativo del generatore che la produce e il simbolo del diodo (*Figura* 5) è stato pensato per evidenziare questo evento; la sua forma a triangolo può essere presa come regola per ricordare il senso di percorrenza della corrente convenzionale. Va solo ricordato che il lato in cui essa entra è detto **anodo** (ed è sottoposto al polo positivo della tensione) mentre quello da cui essa esce è detto **catodo** (ed è collegato al polo negativo della tensione).

Gli **FDN500**)/**FDN507** contengono 8 led all'Arseniuro Fosfuro di Gallio (GaAsP, quindi *rossi*) e in condizioni d'uso corretto assorbono **20mA** con tensione ai loro capi tipica di **1,7V** (**25mA** massimi con tensione applicata di **2V**, pari a **400mW** di dissipazione massima); sono oggetti dal consumo decisamente impegnativo!

Uno dei problemi più sentiti, quando si montano questi dispositivi è quello di ricordare la piedinatura: quando serve d'urgenza, non si riesce mai a trovala...; la *Figura 6* ne riassume l'aspetto, visto da sopra, per i digit a *catodo comune*, ma vale anche per quelli ad *anodo comune*, sostituendo ovviamente (sui pin 3 e 8) la lettera K con la lettera A; non di rado questa piedinatura è mantenuta inalterata anche su modelli alternativi.

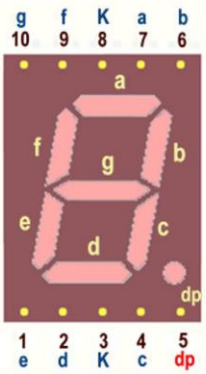


Figura 6 - Single Digit Numeric Display FND500: pinout (top view)

E' interessante notare che la disposizione delle lettere sulla periferia del componente è solo apparentemente illogica: si tratta della sequenza che garantisce il collegamento a ciascuno dei led interni con il percorso più breve possibile, come si può facilmente verificare osservando i dettagli proposti in precedenza.

Anche la disponibilità di 2 catodi (o di 2 anodi, sul pin3 e sul pin8) non deve stupire; basta pensare alla difficoltà nella creazione di circuiti stampati con molti digit: poter disporre di 2 possibilità, di qua o aldilà di numerose fasce di piste, è certamente un vantaggio.

Lo *schema pratico* da utilizzare direttamente nel contesto di un progetto, è proposto dalla *Figura 7a* (per quelli a *catodo comune*, **FDN500**) e in *Figura 7b* (per i digit ad *anodo comune*, **FDN507**).

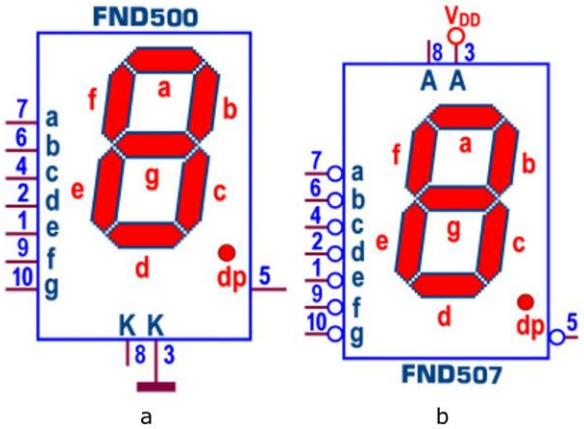


Figura 7 - Single Digit Numeric Display FND500/FND507: schema pratico

Lo *schema funzionale* (*Figura 8*) è naturalmente molto semplice ed evidenzia la *natura a led* di questi dispositivi.

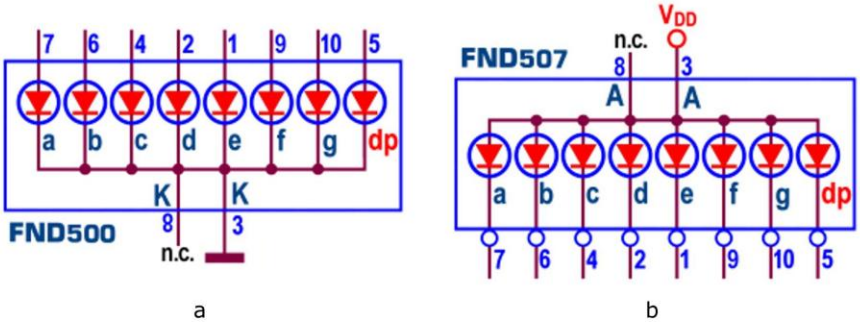


Figura 8 - Single Digit Numeric Display FND500/FND507: schema funzionale

Data la sua simmetria, quando lo utilizziamo realmente può tornare utile sottolineare che uno dei suoi due lati corti è caratterizzato dalla presenza di quattro visibilissime *tacchette* (*Figura 9*): il pin10 di questo componente è il primo a sinistra, visto da sopra, rispetto a questo lato.

Chi dispone di buona vista può arrivare alla stessa conclusione anche individuando il puntino decimale, nell'angolo in basso a destra...

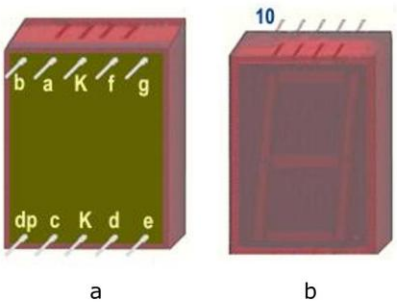


Figura 9 - Single Digit Numeric Display FND500/FND507: aspetto esterno

L'utilizzo diretto di un digit (cioè senza *strato di interfaccia*) è ovviamente legittimo ma non molto ragionevole: se si "brucia" anche solo uno dei led interni tutto il componente è *da buttare*.. Meglio associarlo a dispositivi (componenti logici o porte d'uscita di microcontrollori) in grado di garantirgli le corrette condizioni di carico, sostanzialmente quelle di una batteria di 8 led, alla quali ci si può tranquillamente ricondurre.

Quando il controllo è affidato alle linee d'uscita (*attive alte*) di un dispositivo che lavora in *logica positiva* sono necessari digit a *catodo comune*, come gli **FDN500**; con riferimento alle logiche TTL (*Figura 10*), nel loro funzionamento tipico a livello alto (cioè quando un'uscita è chiamata a pilotare gli ingressi TTL di oggetti simili a se stessa) è obbligatorio assicurarsi che la $\mathbf{V_{OH}}$ non scenda sotto i **2,4V**, il valore limite minimo standard perchè l'uscita stessa venga ritenuta un **1** logico; questa condizione è rispettata purchè la corrente massima erogata non superi i **0,4mA**, più che sufficiente per pilotare una ventina di ingressi TTL LS (condizione nota come *fan-out*).

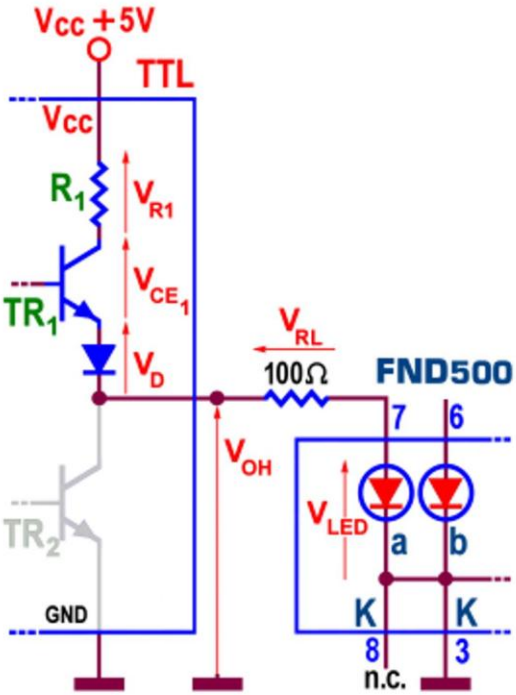


Figura 10 - Numeric Common Catode Display FND500: controllo con logica TTL

Questo non significa che la corrente chiesta all'uscita non possa essere maggiore, ma bisogna essere consapevoli che un'uscita TTL non è adatta (e non è nata) per *erogare corrente*; si può facilmente verificare che maggiore è la corrente richiesta dal carico, minore sarà anche la $\mathbf{V_{OH}}$ disponibile, a causa della maggiore *caduta di tensione interna*, $\mathbf{V_{R1}} + \mathbf{V_{CE1}} + \mathbf{V_D}$; in condizioni standard (cioè con erogazione massima di **0,4mA**) essa è valutata in **2,6V** (dovendone lasciare in uscita $\mathbf{V_{OH}} = \mathbf{2,4V}$, supposta una $\mathbf{V_{CC}}$ di **5V**) e sarà di certo maggiore (anche se difficilmente calcolabile) se la serie $\mathbf{R_1}$, $\mathbf{TR_1}$, **diodo**, sarà percorsa dagli **8mA** desiderati per accendere il Led.

Volendo assicurare un ***resistore*** di limitazione ***in serie*** ad ogni Led (cioè ad ogni *segmento* interno) risulta dunque difficile valutarne la *resistenza*; di certo la tensione residua in uscita si avvicinerà a quella del diodo LED Rosso in zona di funzionamento (da **1,7V** in su) e, in queste condizioni, il *resistore* potrebbe anche non essere necessario; ma, per buona etica, non faremo mancare un minimo di sicurezza, fissando la sua *resistenza* a **100 ohm** (*Figura 11*), probabilmente anche troppo elevata.

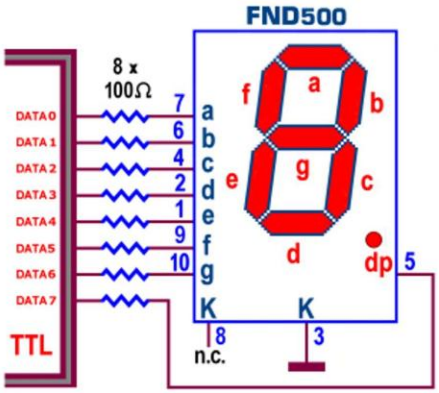


Figura 11 - Numeric Common Catode Display FND500: controllo attivo alto

Nella sua illogicità la scelta di costringere il dispositivo di controllo ad erogare corrente ha un suo vantaggio: non richiede alimentazione esterna e, probabilmente, è questa la ragione della sua diffusione; di sicuro la corrente che l'uscita è costretta ad erogare è di gran lunga maggiore a quella per la quale è stata progettata, anche se questo, nei limiti della decenza, non le nuocerà più di tanto, se non per il fatto di contribuire a riscaldare l'ambiente ...

Dualmente: se il controllo è affidato alle linee logiche di un dispositivo *attivo basso* sono necessari digit ad *anodo comune*, come gli **FDN507**, e il discorso è completamente diverso: ciascuna delle 8 linee d'uscita è chiamata ad assorbire la corrente necessaria all'accensione dei rispettivi led di questa periferica.

Quando la linea d'uscita TTL è a livello basso (**0** logico, *Figura 12*) il transistor $\mathbf{TR_1}$ è interdetto (cioè si comporta come un circuito aperto) mentre il transistor $\mathbf{TR_2}$ è in forte conduzione (cioè *in saturazione*): in questa situazione il valore della tensione d'uscita, $\mathbf{V_{OL}}$, è ora quasi nullo (coincide con la tensione $\mathbf{V_{CE2}}$ di saturazione del transistor $\mathbf{TR_2}$, mai superiore ai **0,3V**) ed è sostanzialmente indipendentemente dalla corrente che è chiamata ad assorbire.

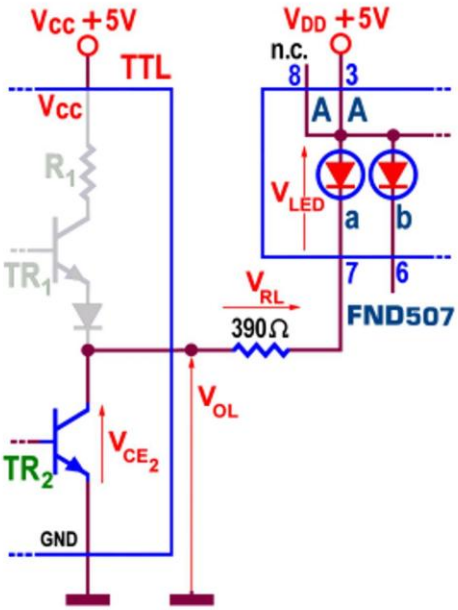


Figura 12 - Numeric Common Anode Display FND507: controllo con logica TTL

Una **situazione ideale** per ogni tipo di applicazione, anche per gestire un digit, i cui led si accenderanno con eccellente luminosità senza alcun tipo di stress o di forzatura per la logica di controllo, mentre se la linea d'uscita è alta (**1** logico) rimarranno spenti, non disponendo ai loro capi di una differenza di potenziale sufficiente a superare la *soglia*.

Il vantaggio di questa soluzione (*Figura 13*) è quello di far lavorare il dispositivo di controllo nel modo ad esso più naturale (essendo intrinsecamente più adatto ad *assorbire* corrente piuttosto che *erogarla*) senza sottoporlo ad alcun surriscaldamento; il valore della resistenza del resistore dovrà assicurare una corrente nei limiti dello standard TTL LS, cioè al massimo di **8mA**.
Il conto è presto fatto: [($\mathbf{V_{CC}}$ - $\mathbf{V_{LED}}$ - $\mathbf{V_{CE2}}$)/$\mathbf{I_{LED}}$] = [(**5V** - **1.7V** - **0.2V**)/**8mA**] = **3,1V**/**8mA** = **387 ohm**, portato al valore normalizzato di **390 ohm**.

Un ulteriore vantaggio sta nella possibilità di impiegare una tensione di alimentazione esterna $\mathbf{V_{DD}}$ diversa da $\mathbf{V_{CC}}$ =**5V**; in questo caso è fondamentale collegare insieme le masse dei 2 alimentatori e ricalcolare il valore della resistenza del resistore (per esempio con $\mathbf{V_{DD}}$ =**12V** esso passerà a **1262 ohm**, normalizzato a **1,2 Kohm**).

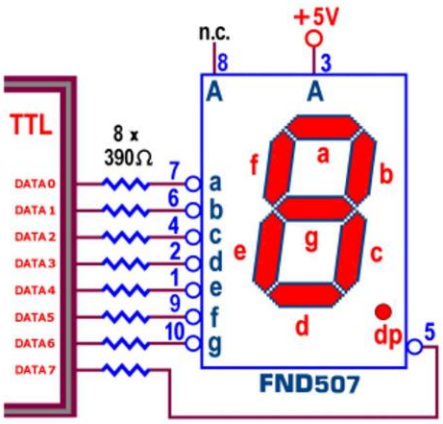


Figura 13 - Numeric Common Anode Display FND507: controllo attivo basso

La situazione ottimale descritta per le logiche di controllo *attive basse* suggerisce una soluzione al problema sollevato per quelle *attive alte*: "bufferizzare" ciascuna delle uscite, cioè farle seguire da un dispositivo (driver) in grado di garantire la corrente necessaria ai led del digit senza "caricarle" oltre misura; ci sono numerosi modi per fare questa operazione (e in parte li ritroveremo anche in alcuni decoder specifici) ma la cosa si può risolvere anche con un semplice transistor che, funzionando da amplificatore di corrente *invertente*, richiederà però l'uso di un **FDN507** invece del **FDN500**.

Siamo giunti al termine di questa trattazione dedicata ai visualizzatori numerici; possiamo ora affrontare con sicurezza i dispositivi TTL e CMOS progettati per interfacciarli; devo peraltro sottolineare che essi non sono strettamente necessari se il controllo è affidato ad un processore o a un microcontrollore.

Entrambi infatti dispongono, direttamente o indirettamente, di porte d'uscita (di solito ad 8 bit) che per certi versi rendono più versatile il controllo di un digit; in primo luogo c'è il vantaggio del controllo assoluto sui caratteri proponibili su di esso: con le opportune combinazioni di *bit attivi* è possibile proporre qualunque simbolo, non solo alfanumerico, consentendo perfino di tenere spento il digit (cosa non sempre possibile con gli integrati specializzati).

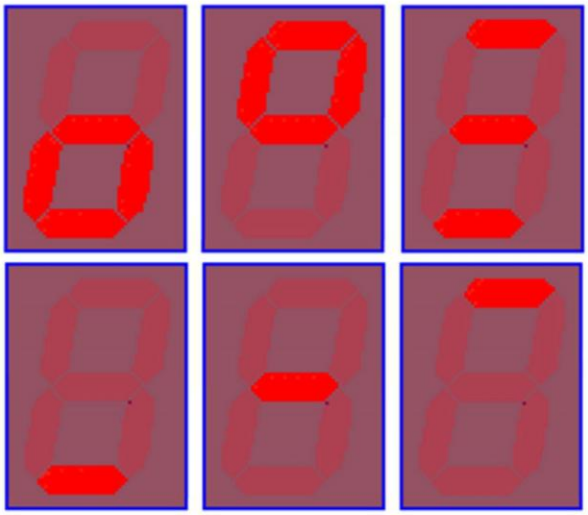


Figura 14 - Single Digit Numeric Display: combinazioni particolari di segmenti

La *Figura 14* mostra alcune di queste combinazioni di segmenti che, sebbene non strettamente necessarie, possono rivelarsi molto utili per fornire particolari segnalazioni, come *allarmi* o *segnali di attesa* e non saranno disponibili con altre soluzioni.

La seconda considerazione riguarda la possibilità di gestire i caratteri alfabetici; un digit a 7 segmenti non è certo il più adatto per questo compito (esistono visualizzatori specificatamente attrezzati per questo scopo), tuttavia con esso possono essere *stilizzate* quasi tutte le lettere.

La possibilità di poterle gestire (di solito con l'aiuto di tabelle di conversione da gestire da software) apre scenari di visualizzazione senza limiti, del tutto impraticabili con l'impiego degli integrati decoder tradizionali.

Decoder per digit a 7 segmenti

Nella precedente premessa abbiamo stabilito che un **digit display** è in grado di mostrare una qualunque delle possibili 256 combinazioni dei suoi led interni (7 segmenti, con o senza il punto decimale); molte di esse sono poco significative in un *contesto alfanumerico*, ma altre possono tornare molto utili.

Di norma però essi sono usati per visualizzare l'informazione binaria fornita da un determinato dispositivo logico (una macchina combinatoria oppure il *Bus Dati* di un microprocessore o quella di una porta di un microcontrollore o della porta parallela di un PC), spesso organizzata su 4 (o multipli di 4) linee d'uscita; di certo, come è facilmente intuibile, nessuno dei possibili *gruppi di 4 bit (nibble*) potrà essere fornito direttamente ad un digit.

Questo problema è stato risolto progettando una circuito in grado di accettare in ingresso un codice binario a 4 bit e rendere disponibili in uscita 7 linee logiche adatte a controllare lo stato (acceso o spento) di altrettanti led.

Il numero binario proposto sulle sue 4 linee d'ingresso potrà essere interpretato (*decodificato*) in parte (solo le prime 10 combinazioni, ***Codice BCD***, per associare un simbolo a ciascuno dei 10 elementi, i numeri da 0 a 9, del *sistema di numerazione decimale*) oppure nella sua interezza (tutte le 16 combinazioni, ***Codice Binario Puro a 4 bit***).

Nel primo caso si parla di **BCD to 7-segment Decoder**: gli integrati disponibili sul mercato sono (quasi) esclusivamente di questo tipo e non c'è da stupirsi, dato che saranno chiamati a formare visualizzatori destinati a mostrare informazioni (numeri..) decimali, come registratori di cassa, bilance, strumenti di ogni tipo,...

Alla seconda categoria appartiene il **9368**; sebbene il suo *datasheet* non preveda per esso una definizione specifica, lo chiamerei **Binary to 7-segment Decoder**, in contrapposizione con quelli della precedente categoria; si tratta di un componente magico, l'unico in grado di mostrare tutti i 16 simboli del *sistema di numerazione esadecimale*, interpretando e decodificando tutte le possibili combinazioni di 4 bit proponibili al suo ingresso, indispensabile per visualizzare (per esempio..) il contenuto dei registri della CPU o di una locazione della memoria.

Va subito evidenziato che sulle 4 linee d'ingresso dei **Decoder da BCD a sette segmenti** è possibile imporre ***tutte*** le possibili 16 combinazioni a 4 bit, ma nessuno di essi è progettato per interpretare le 6 combinazioni più significative, da $(1010)_2=(10)_{10}$ a $(1111)_2=(15)_{10}$; sta al progettista evitare con cura che questo evento accada, altrimenti il digit ad essi collegato fornirà simboli improbabili, di norma inaccettabili.

I valori binari accettabili sono dunque quelli da $(0000)_2$ a $(1001)_2$ che, come abbiamo visto, non solo sono le prime dieci sequenze in *Binario Puro a 4 bit* ma anche le dieci parole del *Codice BCD*; per ciascuna di esse il componente attiverà in uscita i segmenti necessari per creare i corrispondenti 10 simboli del *sistema di numerazione decimale*, da $(0)_{10}$ a $(9)_{10}$, esattamente quelli che verranno mostrati dal digit controllato dal Decoder.

Per questo servizio sono disponibili i componenti **TTL** **74LS47** e **74LS48** (più **7446** e **74LS49**, meno adatti per la gestione diretta di digit) tra loro *funzionalmente identici* e *pin-out compatibili*, ma governati da tecnologie d'uscita diverse, e il componente **CMOS** **4511**, esso pure in parte *pin-out* compatibile (escludendo tre piedini di controllo) con i precedenti ma funzionalmente molto più sofisticato.

Tutti e 4 i citati componenti TTL trattano in modo particolare le possibili 6 combinazioni a 4 bit non appartenenti al codice BCD [da $(1010)_2$ a $(1111)_2$ eventualmente proposte sulle linee d'ingresso] attivando le linee d'uscita di segmento per formare i simboli di *Figura 15.*

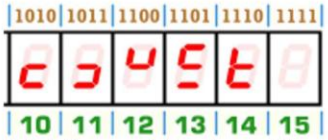


Figura 15 - BCD to 7-segment Decoder: combinazioni non BCD

La disponibilità di questi simboli, piuttosto di altri, ha una logica: chi si è cimentato nel progetto di una macchina combinatoria a partire dalla sua tabella di verità è a conoscenza della disponibilità delle cosiddette *condizioni di indifferenza*; in breve (sarebbe divertente poter andare più a fondo, ma non è questo l'ambito giusto...) partendo dal presupposto che le ultime 6 combinazioni non devono mai essere fornite (perché non appartenenti al codice BCD), il valore logico che la rispettiva uscita può assumere è indifferente, cioè può essere assunto a piacere uguale a **1** o a **0**, nel modo più conveniente ai fini del progetto stesso.

Se, nonostante il divieto, si fornisce (a progetto finito) una delle ultime 6 combinazioni vietate, l'aspetto delle uscite è dunque legato alle scelte imposte alle relative condizioni di indifferenza e si traduce (nel nostro caso) in quello mostrato in *Figura 15*.

Il caso ha voluto che due dei simboli ottenuti siano realistici [una "c" per $(1010)_2$ e una "t" per $(1110)_2$], ma soprattutto l'ultimo [mostrato dal digit con ingressi a $(1111)_2$] si presta ad una interessante considerazione: se il progettista impone in ingresso il codice binario (non BCD) 1111 otterrà come effetto quello di *spegnere il digit* controllato dal Decoder; vedremo che questo effetto, decisamente utile, è gestibile anche da hardware ma, se il dispositivo di

controllo è programmabile (PC o microcontrollore) questa consapevolezza ci consente di passare il codice direttamente da software, con il grande vantaggio di non costare nulla!

La *Figura 16* mostra l'aspetto del simbolo proposto da tutti i componenti TTL per ognuna delle 16 combinazioni possibili; da notare che le scelte imposte dal progetto impongono una inconsueta visualizzazione per il 6 $(0110)_2$ e per il 9 $(1001)_2$, in entrambi i casi priva del trattino in alto o in basso.

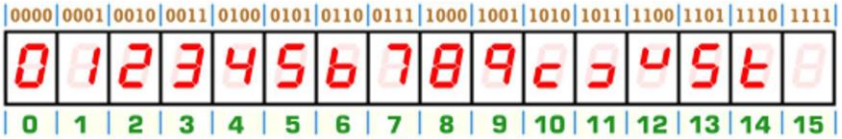


Figura 16 - BCD to 7-segment Decoder: simboli visualizzati per ogni codice d'ingresso

Curioso pensare che i costruttori abbiano preso coscienza del problema perchè la serie di componenti TTL **74LS247**, **74LS248** e **74LS249**, per il resto del tutto identici a quelle con le stesse ultime 2 cifre, mostrano invece il 6 e il 9 nella forma completa del suddetto segmento, rispettivamente "**a**" e "**d**".

L'ingente mole di informazioni disponibili sui **decoder** per *digit a 7 segmenti* impone la necessità di riparlarne, in dettaglio, la prossima puntata.