di GIORGIO OBER (GIOBE2000)

l'angolo di Mr A.KEER

(parte quarta)

PROGETTARE con le PORTE LOGICHE

operatore universale ***NAND***

In questo numero affrontiamo l'analisi dettagliata dei componenti disponibili sul mercato per il primo dei 2 operatori logici universali, noto come NAND, fornendone le caratteristiche, interessanti osservazioni e considerazioni utili alla progettazione.

Quella degli universali è una importante categoria di operatori booleani; la loro universalità consiste nel fatto che, con essi, è possibile realizzare qualunque altro degli operatori logici trattati nelle precedenti puntate (NOT, AND e OR, ma anche OREX e NOREX) e per questa ragione sono numerose le porte logiche destinate a rappresentarli.

OPERATORE NAND (TTL): 74LS00, 74LS10, 74LS20, 74LS30, 74LS133

La serie TTL prevede ben 5 componenti per l'operatore universale **NAND**, a dimostrazione della sua praticità: il **74LS00** contiene 4 porte a 2 ingressi (*Figura 1*); il **74LS10** ne contiene 3 a 3 ingressi (*Figura 2*), il **74LS20** ne contiene 2 a 4 ingressi (*Figura 3*), il **74LS30** e il **74LS133** ne contengono una, rispettivamente a 8 ingressi (*Figura 4*) e a 13 ingressi (*Figura 5*).

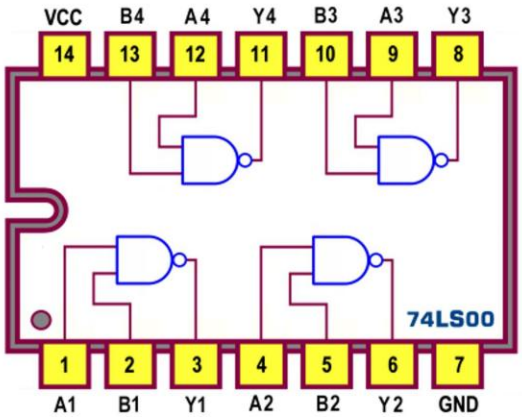


Figura 1 - Quad 2-input NAND Gates 74LS00: Pin-out

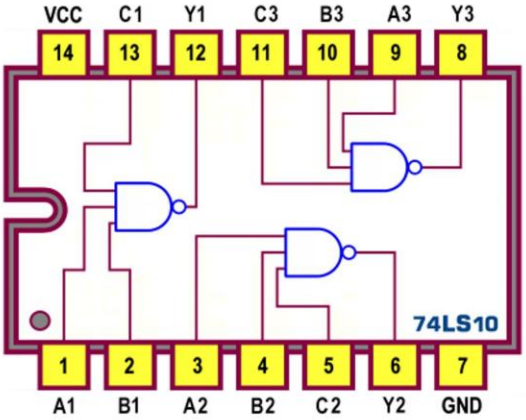


Figura 2 - Triple 3-input NAND Gates 74LS10: Pin-out

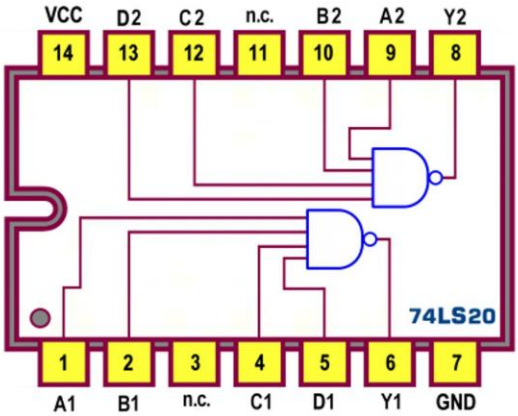


Figura 3 - Dual 4-input NAND Gates 74LS20: Pin-out

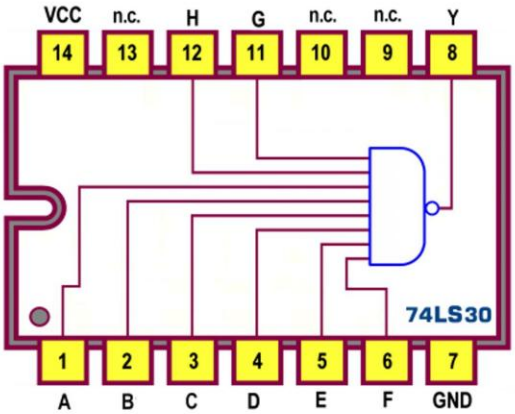


Figura 4 - 8-input NAND Gates 74LS30: Pin-out

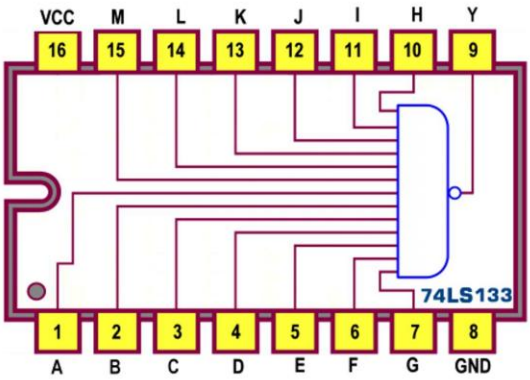
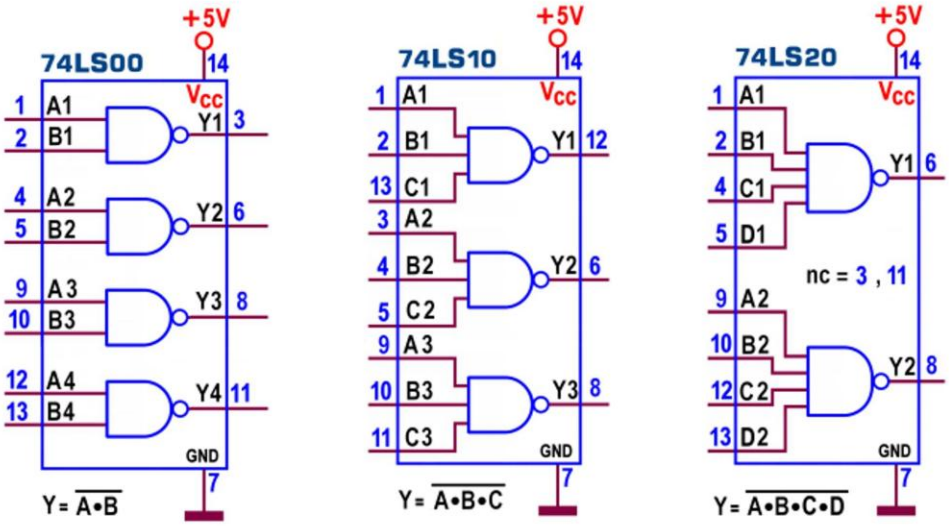


Figura 5 - 13-input NAND Gates 74LS133: Pin-out

Il nome di questo operatore (e delle porte ad esso associate, descritte dai *datasheet* come **NAND Gates**) deriva dalla sua natura di *AND negata*, **N**OT-**AND**, e mette in evidenza la funzione *prodotto logico negato* che è chiamato a realizzare.

Per ciascuno dei 5 componenti la *Figura 6* mostra lo *schema funzionale*, con il *numero* e la *funzione* affidata dal costruttore a ciascun piedino; esso è simile allo *schema logico* (*Logic Diagram*) talvolta offerto al suo posto da alcuni *datasheet*.



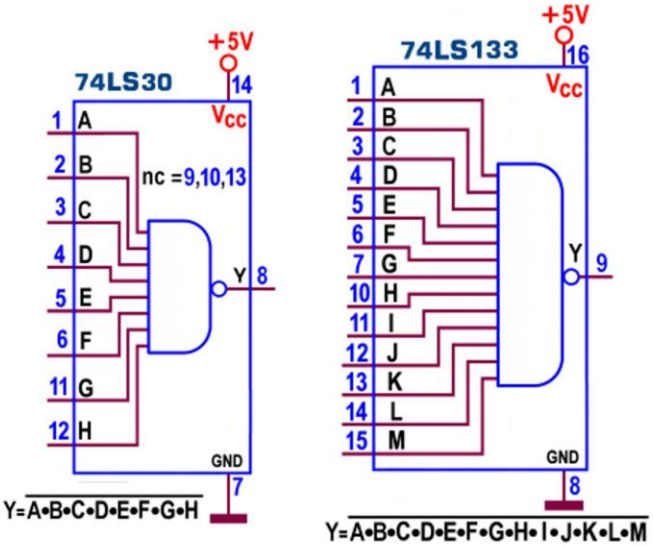


Figura 6 - NAND Gates [TTL]: Schema funzionale

Salta subito agli occhi la disponibilità di porte *con molti ingressi*, non prevista per nessun operatore fondamentale, ma indispensabile nel progetto dei computer: ad esse è spesso affidata, per esempio, la decodifica degli indirizzi del microprocessore, al fine di localizzare con relativa facilità una sua specifica *periferica*.

Senza entrare nel merito (ne riparleremo in una prossima occasione..) quando il processore decide che è il momento per *dare o ricevere dati* da un determinato chip di *memoria* o da un ben preciso *dispositivo di Input/Output*, mette per un breve istante il suo indirizzo sul *bus address* (una struttura multifilare in uscita da esso stesso).

La *Figura 7* mostra l'azione imposta su due **74LS30** dal valore numerico collocato dalla CPU sul *bus address*; è facile intuire che, se il numero che esprime l'indirizzo è (come nel nostro caso) un numero binario della dimensione di un byte, esistono 2^8=256 diversi modi per esprimerlo, da $(00000000)_2$ a $(11111111)_2$.

Va detto che la *Figura 7* offre una visione semplificata delle azioni necessarie alla CPU per contattare un suo dispositivo (simulato nello schema dalla presenza di un led): la porta è indispensabile per identificare con certezza *uno e uno solo* dei possibili indirizzi predisposti dalla CPU sul suo *bus*, ma la reale *rete di selezione* (della quale la **NAND** è parte integrante) dovrà coinvolgere anche altri segnali della CPU stessa!

Dovendo assicurare alle porte **74LS30** il rispetto della loro logica **NAND**, è facile constatare che l'uscita (*attiva bassa*) di ciascuna sarà vera (**0** logico) solo se *tutti* i suoi ingressi (*attivi alti*) sono veri (**1** logico): la prima, collegata al *bus* in modo diretto, attiverà l'uscita solo in presenza della parola binaria 11111111, l'unica (delle 256) a *produrre effetto* sulla sua uscita; qualunque altra combinazione di bit sarà ignorata da questa **NAND**.

Per consentire alla *rete di selezione* di ciascun dispositivo il riconoscimento della propria parola a 8 bit (diversa da quella degli altri) è dunque necessario intervenire aggiungendo uno o più inverter in ingresso alle rispettive **NAND**.

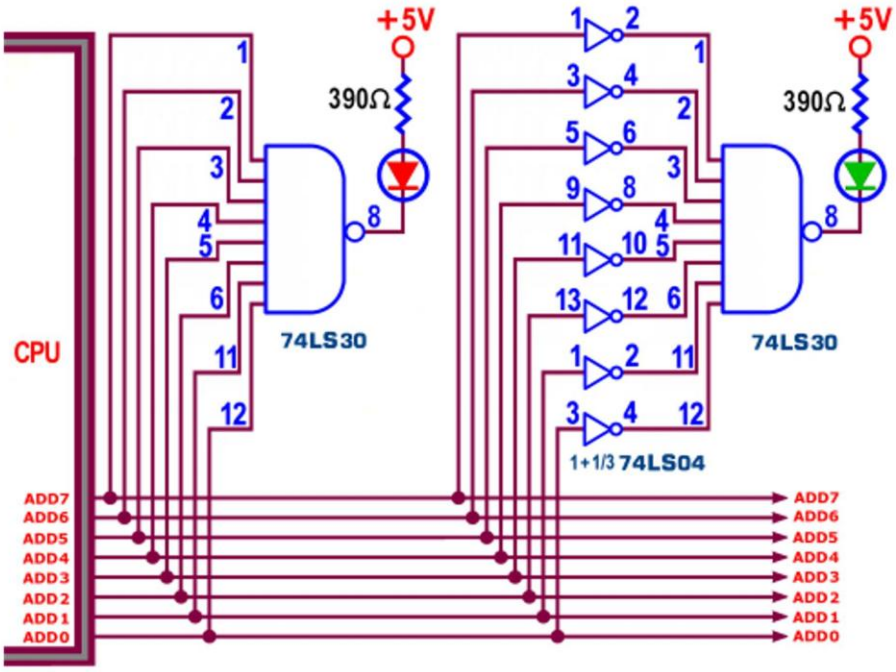
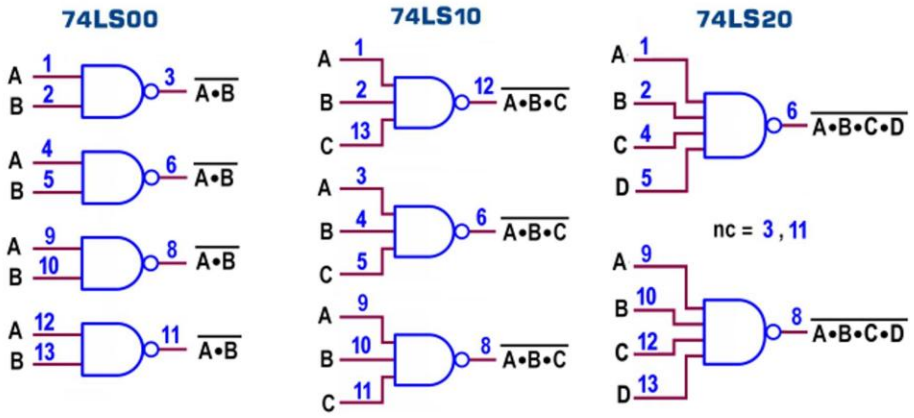


Figura 7 - NAND Gates [TTL]: Selezione dispositivi

Così la seconda **NAND** di *Figura 7*, disponendo di un inverter su ciascuno dei suoi ingressi, attiverà la sua uscita solo in presenza (sul *bus*) dell'indirizzo 00000000, l'unico ad assicurare (sugli ingressi effettivi) la parola binaria 11111111 prevista dalla sua logica di funzionamento.

Lo *schema pratico* (*Figura 8*) raccoglie, per ogni componente, il simbolo delle singole porte, da utilizzare direttamente nel contesto di un progetto, evidenziando il nome dei *segnali logici* che entrano e di quelli che escono e il *numero* dei piedini ad essi riservati.



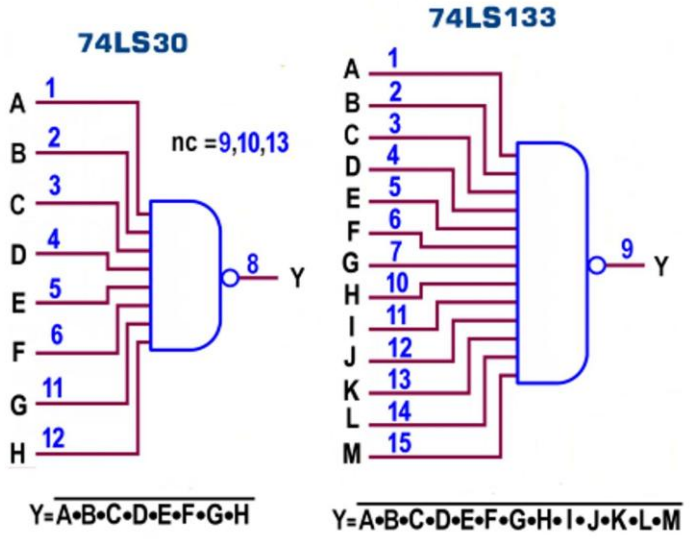
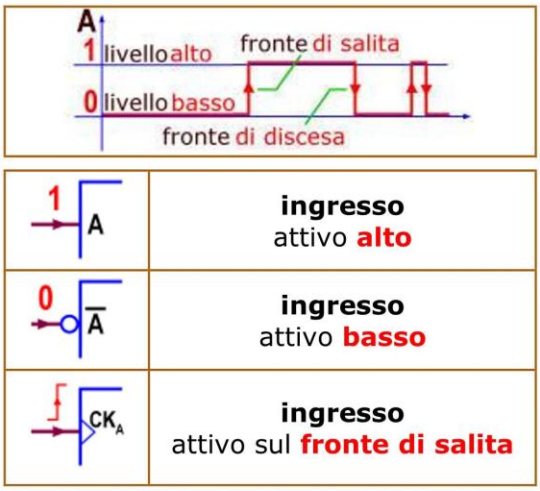


Figura 8 - NAND Gates [TTL]: Schema pratico

Dall'attenta osservazione degli schemi proposti appare evidente la presenza del simbolo delle porte **AND**, mentre quella dell'inverter è sintetizzata dal ***pallino*** posto sull'uscita; si tratta di una buona occasione per sottolineare la tecnica usata per *rappresentare graficamente* il modo con cui possono operare i *segnali d'ingresso* e *d'uscita* di un dispositivo logico.

Fermo restando il *buon gusto*, talvolta disatteso, di porre a sinistra i segnali entranti e a destra quelli in uscita, le variabili chiamate in causa in entrambe le situazioni possono essere ricondotte ad un segnale capace di assumere nel tempo entrambi i 2 *livelli logici alto* **H** e *basso* **L**, da intendersi rispettivamente **1** e **0** in *logica positiva* oppure **0** e **1** in *logica negativa*.

La *Figura 9* mostra la forma di un tipico *segnale logico* e la *simbologia* (caldamente consigliata) da utilizzare negli schemi per rendere immediatamente chiara, nel funzionamento di un circuito logico, la parte *attiva* del segnale coinvolto.



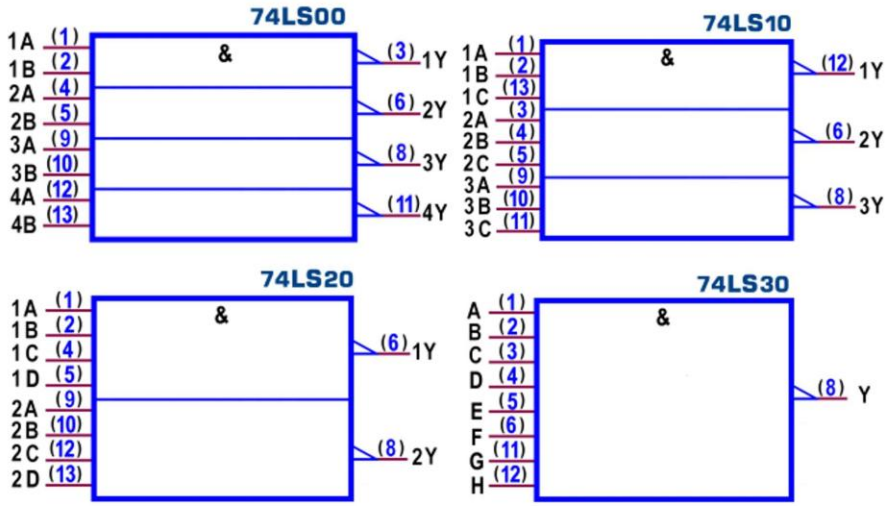
CK$_A$	**ingresso** attivo sul **fronte di discesa**
Y 1	**uscita** attiva **alta**
$\overline{Y}$ 0	**uscita** attiva **bassa**

Figura 9 - TABELLA delle simbologie per gli ingressi logici

Curioso sottolineare che di un segnale logico *non si butta nulla*; i 2 *livelli alto* **H** e *basso* **L** ne sono la parte più eclatante ma, come vedremo, non sempre quella utile al funzionamento: molti dispositivi (i più "intelligenti") lavorano solo in corrispondenza di una delle 2 possibili transizioni tra un livello e l'altro, sinteticamente definita *fronte attivo*, *di salita* (da **L** a **H**) o *di discesa* (da **H** a **L**).

La durata di questa transizione dipende dalla natura del circuito che l'ha generata ma è sempre dell'ordine delle decine di ns.

La *Figura 10* raccoglie il *simbolo logico* di ciascuno dei 5 componenti: pur essendo poco intuitivo e molto diverso da quello utilizzato tradizionalmente esso è riconosciuto come *standard* dalle norme internazionali; in essi la funzione del *pallino* d'uscita è assolta da un *triangolino*...



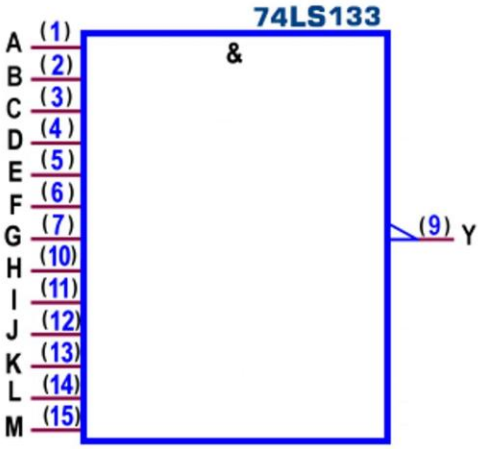


Figura 10 - NAND Gates [TTL]: Simbolo logico ANSI/IEEE Std. 91-1984

La *Tabella di Verità* (*Function Table*) dell'oggetto presente nell'integrato non manca mai nei *datasheet*; la *Figura 11* propone quella di ciascuna delle porte contenute in 3 dei 5 componenti sotto esame.

74LS00

INPUT		OUTPUT
A	B	Y
L	X	H
X	L	H
H	H	L

74LS10

INPUT			OUTPUT
A	B	C	Y
L	X	X	H
X	L	X	H
X	X	L	H
H	H	H	L

74LS20

INPUT				OUTPUT
A	B	C	D	Y
L	X	X	X	H
X	L	X	X	H
X	X	L	X	H
X	X	X	L	H
H	H	H	H	L

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

Figura 11 - NAND Gates [TTL]: Tabella di verità

Quella dei rimanenti 2, decisamente ingombrante per la presenza di 8 e 13 ingressi, è facilmente estrapolabile osservando la simmetria con cui sono incasellate le lettere **H** e **L** (in rappresentanza dei *livelli logici alto* e *basso*) e le lettere **X** (in vece della *condizione di indifferenza*, rappresentativa a piacere o dell'uno o dell'altro, *Either Low or High Logic Level*).

La sintetica *Tabella di verità* di *Figura 11* può quindi essere riproposta esplicitamente in *logica positiva* (**H**=**1**=5V e **L**=**0**=0V, come quella tradizionale) o in *logica negativa* (**H**=**0**=0V e **L**=**1**=5V); la *Figura 12*, relativa ad ogni *porta logica* contenuta nel **74LS00**, sottolinea come essa si comporti come *operatore* **NAND** in *logica positiva* e come *operatore* **NOR** in *logica negativa*.

74LS00

tabella GENERICA

INPUT		OUTPUT
A	B	Y
L	X	H
X	L	H
H	H	L

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

in logica POSITIVA

INPUT		OUTPUT
A	B	Y
0	x	1
x	0	1
1	1	0

1 livello logico **alto**
0 livello logico **basso**

in logica NEGATIVA

INPUT		OUTPUT
A	B	Y
1	x	0
x	1	0
0	0	1

0 livello logico **alto**
1 livello logico **basso**

X o livello logico **alto** o livello logico **basso**

INPUT		OUTPUT
A	B	Y = NAND
0	0	1
0	1	1
1	0	1
1	1	0

$f(A,B) = \overline{A*B}$

INPUT		OUTPUT
A	B	Y = NOR
0	0	1
0	1	0
1	0	0
1	1	0

$f(A,B) = \overline{A+B}$

Figura 12 - NAND Gates 74LS00: Tabella di verità in logica positiva e negativa

L'operatore **NAND** (come quello **NOR**) è ritenuto *universale* per la sua capacità di *trasformarsi* in un qualunque altro operatore. In particolare è molto facile trasformare una **NAND** in **NOT**: basta unire tra loro i piedini della porta di uno qualunque dei componenti trattati in questa presentazione; osservando la *Figura 13a* è facile verificare che, se in ingresso c'è **1** in uscita avremo $\overline{1*1} = \overline{1} =$**0** e se c'è **0** avremo invece $\overline{0*0} = \overline{0} =$**1**; questa situazione può sembrare quantomeno curiosa ma, in realtà, offre una grossa opportunità.

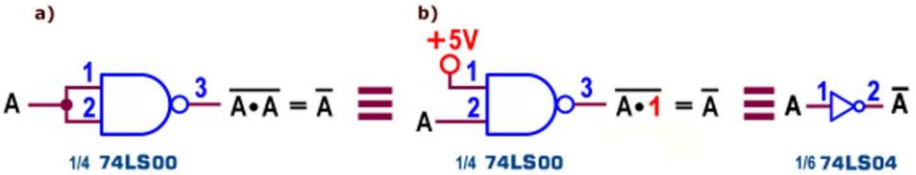


Figura 13 - NAND Gates: Realizzare un INVERTER

Non di rado può succedere di aver a disposizione una o più porte **NAND**, perchè non necessarie al progetto logico e comunque assicurate dai componenti coinvolti; con questa tecnica è possibile disporre di tanti inverter quante sono le porte **NAND** inutilizzate, senza dover ricorrere all'integrato specifico (come il **74LS04**) che li renda disponibili, con evidente risparmio.

Ragionando su questa esigenza abbiamo occasione di sottolineare un altro fatto importante, tipico dei dispositivi logici, in generale: non è mai opportuno *lasciare scollegati* gli ingressi inutilizzati ed è sempre saggio collegarli ad un livello basso (GND=0V) o alto (la tensione d'alimentazione V_{CC}=5V, magari con l'aiuto di un resistore (*pull-up resistor*); in aggiunta, gli ingressi delle logiche **TTL**, se lasciati aperti, *sentono un **1*** (cioè si portano allo stato **H**) e questo, pur non comportando alcun danno al componente, può portare a commutazioni indesiderate dovute a un qualunque rumore *accoppiato capacitivamente* al piedino libero (cioè rilevato per la presenza della sua capacità parassita).

Quanto detto può essere preso in considerazione anche per trasformare una **NAND** in **NOT** (o in qualunque altro operatore, come vedremo): alcuni suoi piedini possono essere resi ininfluenti collegandoli a un livello opportuno; la *Figura 13b* ci aiuta a capire che, per *tenere aperta* la porta, dovremo collegarli a **1** (facile verificare che se, per errore, avessimo messo

una massa (0V) l'uscita della **NAND** sarebbe stata forzata comunque a **1**, indipendentemente dal valore della variabile posta sull'altro piedino, precludendone dunque il passaggio).

Costruire una **AND** con sole porte **NAND** è una banalità: è sufficiente utilizzarne 2 in cascata (*Figura 14*) usando la seconda come inverter; la rete è giustificata dalle regole di Boole, una delle quali afferma che la *doppia negazione* equivale ad una *affermazione* (cioè, in sostanza, la *toglie di mezzo*..)

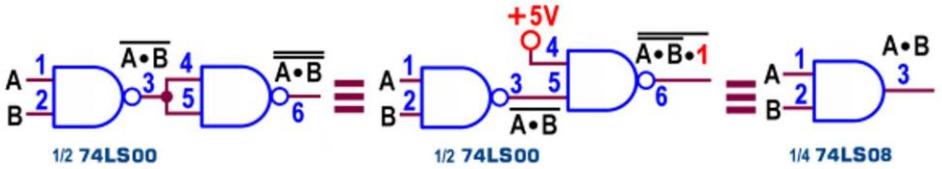


Figura 14 - NAND Gates: Realizzare una AND

Meno immediato è capire come si possa realizzare una **OR** con sole porte **NAND**: è necessario far riferimento ancora ai principi e alle regole dell'algebra di Boole; una delle più importanti (la chiave per passare dalla *logica positiva* alla *logica negativa*) è il *teorema di De Morgan* che, da solo (*Figura 15a*) mostra la via da seguire per arrivare al risultato desiderato (*Figura 15b*): la porta **OR** si ottiene utilizzando una **NAND** con ingressi negati.

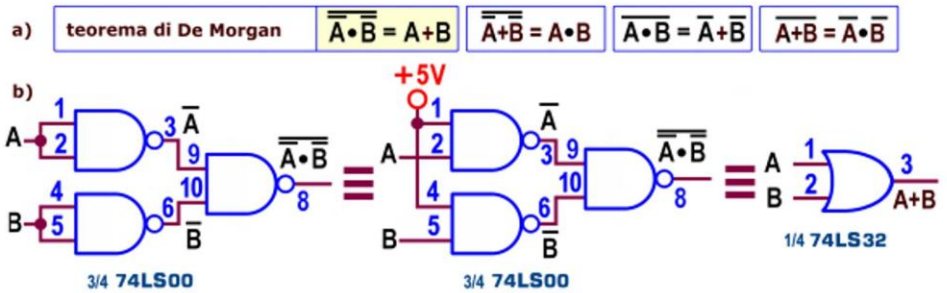


Figura 15 - NAND Gates: Realizzare una OR

L'aggiunta di una ulteriore porta con piedini collegati tra loro (cioè con le funzioni di inverter) è la ovvia soluzione per realizzare **NOR** con le **NAND**.

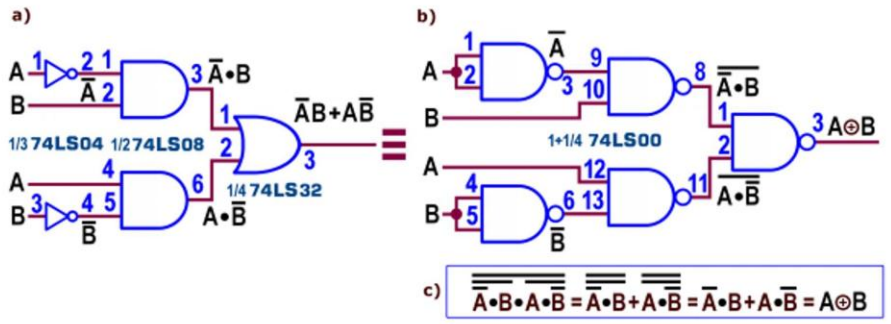
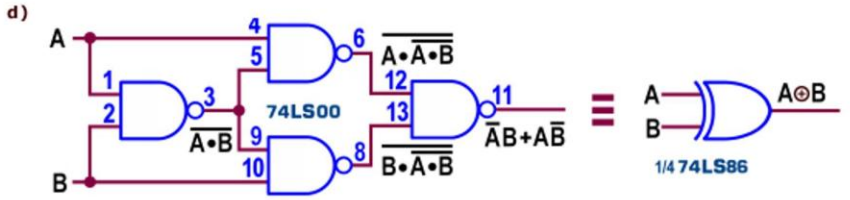


Figura 16 - NAND Gates: Realizzare una OR ESCLUSIVO

Un discorso a parte merita l'analisi del metodo per ottenere un **OR ESCLUSIVO** con sole porte **NAND**: è ragionevole far riferimento allo schema suggerito la scorsa puntata, ottenuto dalla sintesi dell'equazione $\overline{\mathbf{A}}\,\mathbf{B}+\mathbf{A}\,\overline{\mathbf{B}}$ per essa definita dalla teoria booleana e riproposto nella *Figura 16a*; sostituendo le porte **NOT**, **AND** e **OR** con le rispettive equivalenze **NAND** si ottiene il circuito di *Figura 16b*, la cui valenza è dimostrata analiticamente dalla formula della *Figura 16c*. Risulta evidente una fastidiosa realtà: servono 5 **NAND** a 2 ingressi, una in più di quante disponibili del componente **74LS00**.

Forse per questa ragione un geniale pensatore (di cui non conosco il nome ma al quale va tutta la mia ammirazione) si è *spremuto* fino al punto di realizzare un **OR ESCLUSIVO** con sole 4 **NAND** a 2 ingressi (*Figura 16d*), cioè al costo di un solo integrato!! La *Figura 16e* mostra i laboriosi passaggi algebrici in grado di dimostrare la validità di questo progetto: il tutto va per altro inteso come una evidente sfida intellettuale, ben sapendo che lo stesso operatore è presente in ben 4 copie nel **74LS86**...



e) $\overline{\overline{A\bullet\overline{A\bullet B}}\bullet\overline{B\bullet\overline{A\bullet B}}} = \overline{\overline{A\bullet\overline{A\bullet B}}} + \overline{\overline{B\bullet\overline{A\bullet B}}} = A\bullet\overline{A\bullet B} + B\bullet\overline{A\bullet B} = A\bullet(\overline{A}+\overline{B})+B\bullet(\overline{A}+\overline{B}) =$
$= A\bullet\overline{A}+A\bullet\overline{B}+B\bullet\overline{A}+B\bullet\overline{B} = \overline{A}\bullet B+A\bullet\overline{B} = A\oplus B$

Figura 16 (segue) - NAND Gates: Realizzare una OR ESCLUSIVO

In chiusura possiamo soffermarci su un ultimo importante aspetto concreto; abbiamo già sottolineato la disponibilità di porte con una *smisurata* quantità di ingressi: esistono versioni a 2, a 3 o a 4 ingressi, come per altri operatori, ma anche a 8 e a 13!! In questo caso, però, si tratta di una cosa ben pensata non essendo possibile *porre in cascata* più *porte* **NAND**, poichè esse non godono più della proprietà associativa.

La *Figura 17a* mette in evidenza che 2 porte **NAND** a 2 ingressi *in cascata* **non** realizzano una **NAND** a 3 ingressi, mentre questa esigenza è ottenuta con l'aggiunta di un inverter tra le 2 (*Figura 17b*), naturalmente al costo di una ulteriore porta..

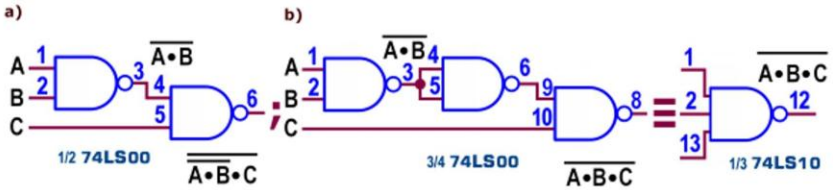


Figura 17 - NAND Gates: a 3 ingressi in cascata

Le caratteristiche elettriche delle *porte* **NAND** trattate finora sono quelle della *famiglia logica* **TTL LS**, proposte in dettaglio la prima puntata; tra esse è utile ricordare la corrente massima *assorbita* in uscita a livello **0**, $\mathbf{I_{OL}}$=**8 mA**, e la tensione *tipica* in uscita a livello **1**, $\mathbf{V_{OH}}$=**3,5 V**.

La potenza dissipata massima (con uscite a **0**) decresce all'aumentare degli ingressi disponibili, da **22 mW** (**74LS00**) a **5,5 mW** (**74LS133**) mentre il *ritardo di propagazione* massimo (*Propagation Delay Time*, con carico di **2k**ohm/**15**pF) per entrambe le transizioni ($\mathbf{t_{PLH}}$ e $\mathbf{t_{PHL}}$) è per tutti pari a **10 ns**, con eccezione del **74LS133**, per il quale vale **59ns**.

OPERATORE NAND (CMOS): 4011, 4023, 4012, 4068

La serie CMOS riserva 4 componenti all'operatore universale **NAND**: il **4011** contiene 4 porte a 2 ingressi (*Figura 18*); il **4023** ne contiene 3 a 3 ingressi (*Figura 19*), il **4012** ne contiene 2 a 4 ingressi (*Figura 20*) e il **4068** ne contiene 1 a 8 ingressi (*Figura 21*); curiosamente questo componente offre una seconda uscita sulla quale rende disponibile anche la funzione **AND** a 8 ingressi.

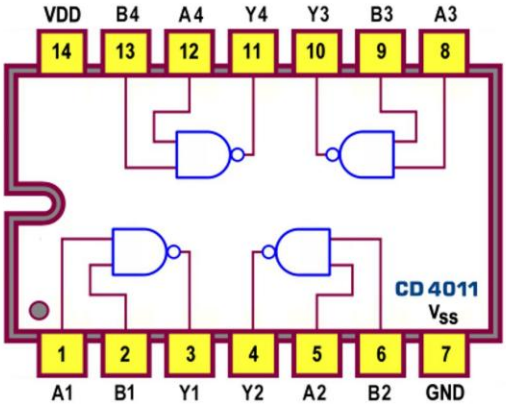


Figura 18 - Quad 2-input NAND Gates 4011: Pin-out

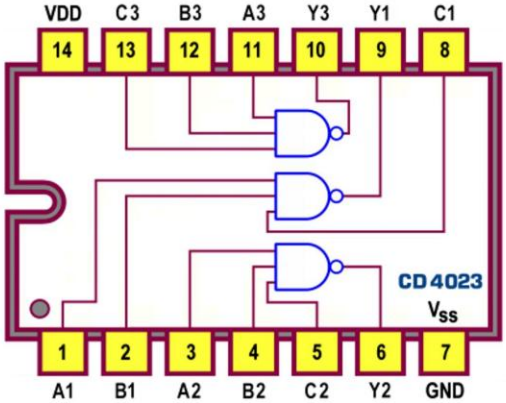


Figura 19 - Triple 3-input NAND Gates 4023: Pin-out

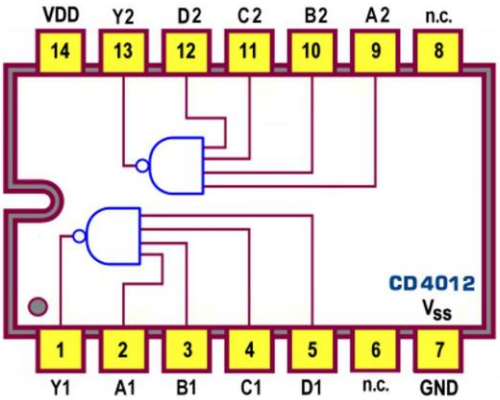


Figura 20 - Dual 4-input NAND Gates 4012: Pin-out

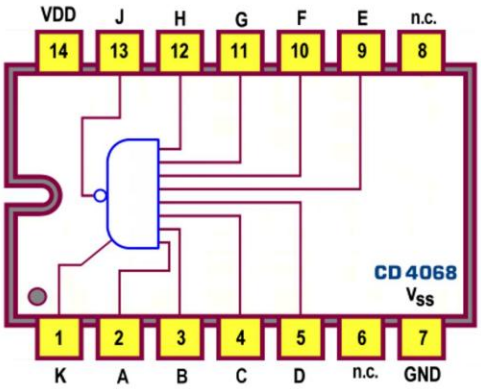


Figura 21 - 8-input NAND Gates 4068: Pin-out

Anche per questi 4 componenti sono disponibili gli *schemi funzionali* e *pratici*, mostrati rispettivamente in *Figura 22* e in *Figura 23*.

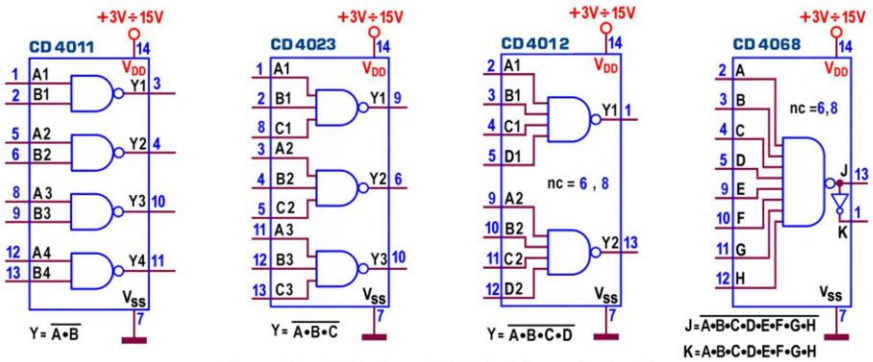


Figura 22 - NAND Gates [CMOS]: Schema funzionale

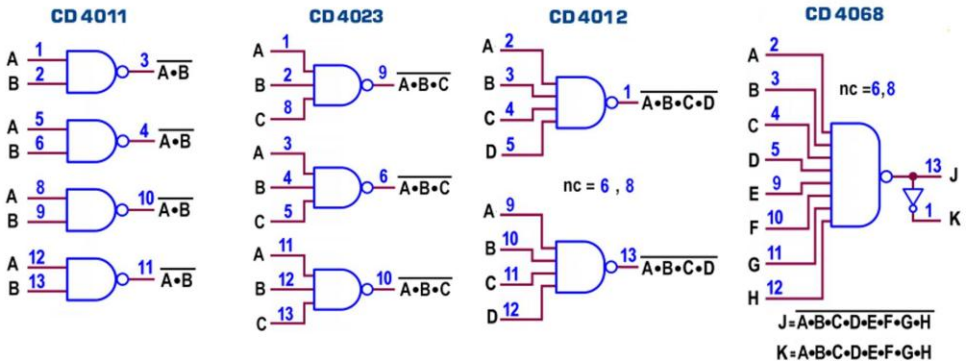


Figura 23 - NAND Gates [CMOS]: Schema pratico

Le caratteristiche elettriche dei 4 componenti sono quelle della *famiglia logica* **CMOS**, proposte in dettaglio la prima puntata; in sintesi: la tensione di alimentazione $\mathbf{V_{DD}}$ può variare da **3V** a **15V**; le uscite assicurano lo stesso valore sia per la corrente *assorbita* $\mathbf{I_{OL}}$ a livello **0** che per quella *erogata* $\mathbf{I_{OH}}$ a livello **1**, da **1mA** (con $\mathbf{V_{DD}}$=5V) fino a **8mA** (con $\mathbf{V_{DD}}$=15V); i livelli di tensione su ogni uscita sono tipicamente uguali alla $\mathbf{V_{DD}}$ per la $\mathbf{V_{OH}}$ (a livello alto) e alla $\mathbf{V_{SS}}$ (=0V) per la $\mathbf{V_{OL}}$ (a livello basso); la *potenza dissipata* è trascurabile (qualche **µW**); il *ritardo di propagazione* massimo $\mathbf{t_{PLH}}$ e $\mathbf{t_{PHL}}$ (con carico di **200k**ohm/**50**pF) varia mediamente da **250 ns** a **70 ns**, al crescere del valore dell'alimentazione.