

**Lezioni di:
Laboratorio di elettronica digitale**

Anno Accademico 2006-2007

ELETRONICA DIGITALE

Laboratorio:	lunedì	14:30-17:30
	martedì	8:30-11:30
Lezioni:	giovedì	14:30-17:30

Corso di Laurea in Fisica e Astrofisica - a.a. 2006/07

SECONDO ANNO

III TRIMESTRE		Lunedì		martedì		Mercoledì		giovedì		Venerdì	
ONDE ELETTR. E OTTICA (con meteorologia e sensoristica ambientale, II anno)	Dalpiaz	9.00-11.00	Catt/19	11.30-13.30	Catt/19			11.00 – 13.00	Catt/19		
LABORATORIO DI ELETTRONICA DIGITALE	Savrié	14.30-17.30	Lab. elettronica – F3	8.30-11.30	Lab. elettronica			14.30-17.30	Lab. elettronica		
STUD.FUNZ.INT.FIS. (con meteorologia e sensoristica ambientale, II anno)	Comelli	11.00 – 13.00	Catt/19			8.30-10.30	Catt/19	9.00-11.00	Catt/19		

occupazione laboratorio elettronica edificio F

III TRIMESTRE	Lunedì	martedì	Mercoledì	giovedì	Venerdì
LABORATORIO DI ELETTRONICA					
Savrié	14:30-17:30	11.30-13.30		14.30 – 17.30	
Marziani		14.30-17.30	10.30-13.30		11.30– 13.30
Andreotti			forse		8:30-11:30

•ELETTRONICA

perché ci interessa?

SISTEMI TIPICI

sistema informatico
impianti HIFI

SEGNALI

analogici
digitali

•APPROCCIO SISTEMISTICO

sistemi

→→ sistema per la misura della velocità del suono

apparati

→→ C.R.O.

blocchi funzionali

→→ amplificatore, trigger di Schmidt, alimentatori.....

schemi circuitali

→→ molto complessi

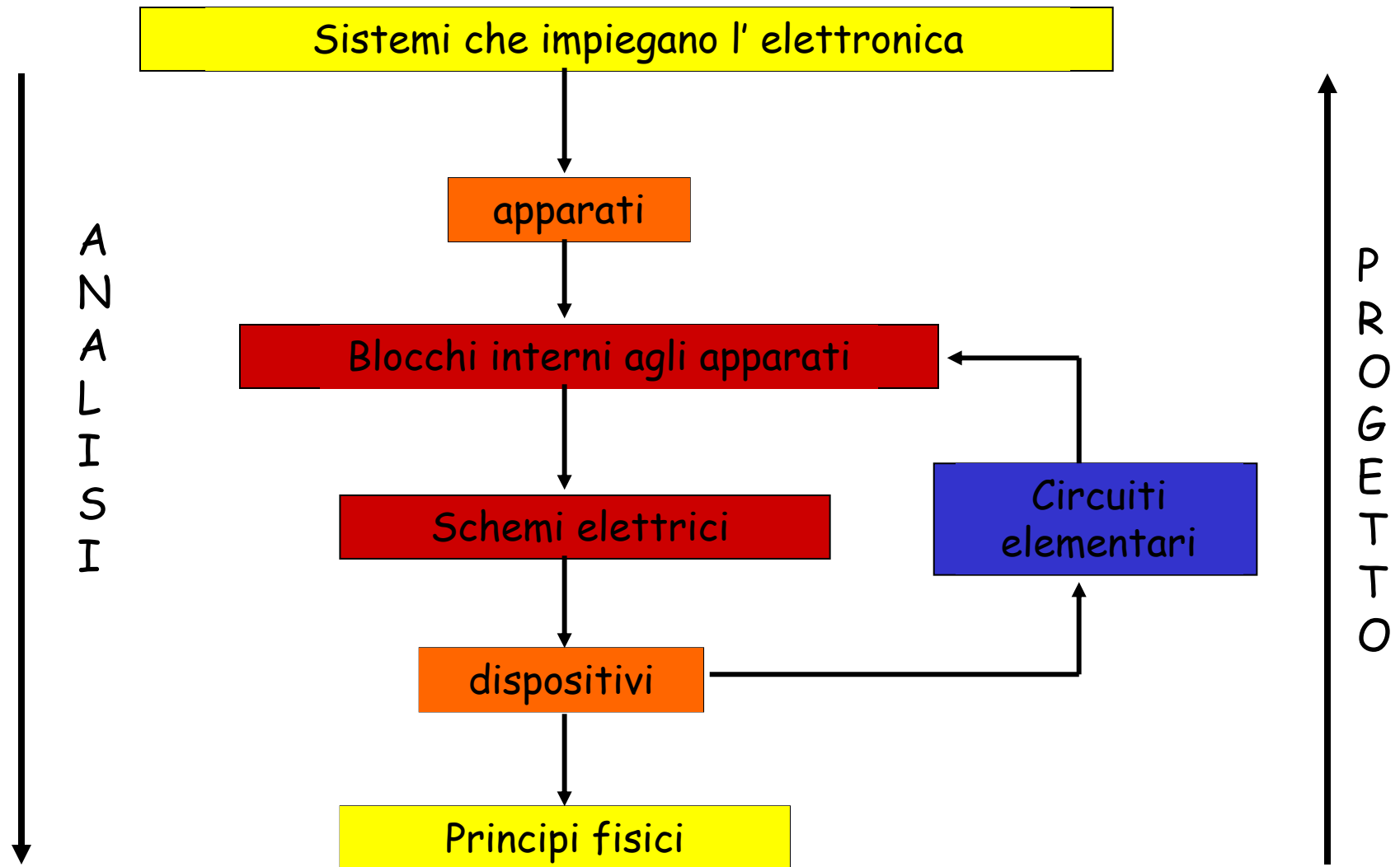
componenti

→→ molto pochi e ricorrenti: circ. integrati e componenti

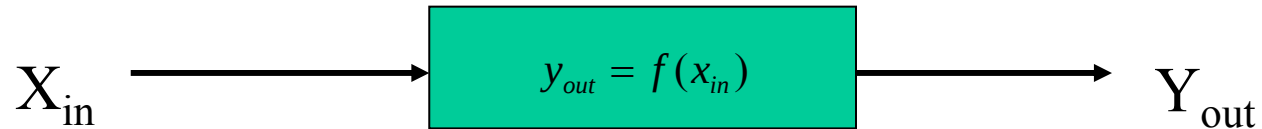
Riduce tutto, a qualunque livello, al concetto di blocco funzionale:

radio,TV, stereo, strumentazione varia.....

Inserire un esempio di Circuito complesso



IL BLOCCO FUNZIONALE



Proprietà generali dei blocchi:

È completamente determinato dalla funzione che lega le variabili di ingresso e di uscita

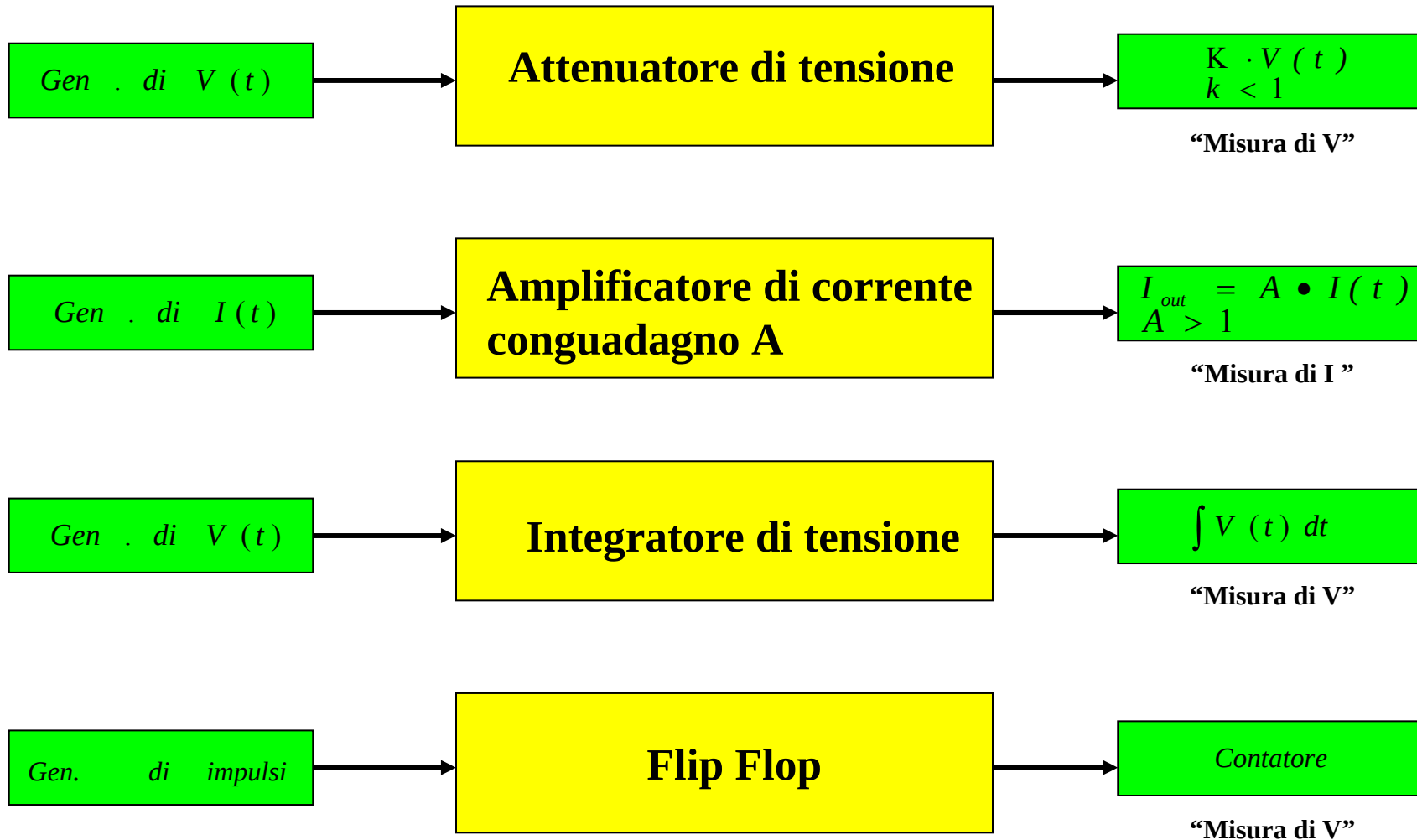
Possono essere:

Digitali	:	elettronica digitale
Analogici	:	elettronica analogica
Di conversione	:	A/D & D/A

Possono anche essere:

Lineari e non lineari

ESEMPI



PERCHE' APPROCCIO SISTEMISTICO?

- sistema = blocco
- blocco di natura elettronica

PERCHE' PRIMA L' ELETTRONICA DIGITALE?

- più facile
- non richiede nozioni preliminari
- candidato ideale al tipo di approccio
- due soli stati (variabili di ingresso: tensioni)
fisici: H,L
logici: T,F; 1,0; ... sec. i gusti

- famiglie logiche:

TTL, HTL,ECL,
MOS,CMOS....

- logica

positiva: $H \longrightarrow T/1$, $L \longrightarrow F/0$

negativa: $L \longrightarrow T/1$, $H \longrightarrow F/0$

- funzioni logiche

le stesse per tutte

Family	Basic gate	Fanout	Pd (mW/gate)	Noise immunity	Prop. delay (ns/gate)	Clock (MHz)
TTL	NAND	10	10	VG	10	35
TTL-H	NAND	10	22	VG	6	50
TTL-L	NAND	20	1	VG	33	3
TTL-LS	NAND	20	2	VG	9.5	45
TTL-S	NAND	10	19	VG	3	125
TTL-AS	NAND	40	10	VG	1.5	175
TTL-ALS	NAND	20	1	VG	4	50
ECL 10K	OR-NOR	25	40-55	P	2	>60
ECL100K	OR-NOR	??	40-55	P	0.75	600
MOS	NAND	20	0.2-10	G	300	2
74C	NOR/NAND	50	0.01/1	VG	70	10
74HC	NOR/NAND	20	0.0025/0.6	VG	18	60
74HCT	NOR/NAND	20	0.0025/0.6	VG	18	60
74AC	NOR/NAND	50	0.005/0.75	VG	5.25	100
74ACT	NOR/NAND	50	0.005/0.75	VG	4.75	100

- la logica usata
speculazione “intellettuale del
XIX secolo:

ALGEBRA DI BOOLE

- costanti: 0,1; T,F; H,L.....
- variabili: x,y,z.... ma ognuna ha 2 soli valori!
- funzioni: $f(x,y,.....)$...come sopra
- solo 3 operazioni (fondamentali):

NOT agisce solo su 1 var, cost. o funzione

AND agisce su 2 o più var, cost. o funzioni

OR agisce su 2 o più var, cost. o funzioni

POSTULATI....

$$1) A = 0 \text{ o } A = 1$$

$$2) 0 \bullet 0 = 0$$

$$3) 1 \bullet 1 = 1$$

$$4) 1 \bullet 0 = 0 \bullet 1 = 0$$

$$5) 0 + 0 = 0$$

$$6) 1 + 0 = 0 + 1 = 1$$

$$7) 1 + 1 = 1$$

....& TEOREMI

$$1) A + B = B + A \quad A \bullet B = B \bullet A$$

$$2) (A + B) + C = A + (B + C)$$

$$(A \bullet B) \bullet C = A \bullet (B \bullet C)$$

$$3) A \bullet (B + C) = A \bullet B + A \bullet C$$

$$A + (B \bullet C) = (A + B) \bullet (A + C)$$

$$4) A + A = A \quad A \bullet A = A$$

$$5) \overline{\overline{A}} = A$$

$$\begin{aligned}
6) \quad & A + (A \cdot B) = A \\
& A \cdot (A + B) = A \\
7) \quad & 0 + A = 1 \cdot A = A \\
& 1 + A = 1 \quad 0 \cdot A = 0 \\
8) \quad & \overline{A} + A = 1 \quad \overline{A} \cdot A = 0 \\
9) \quad & A + \overline{A} B = A + B \\
& A \cdot (\overline{A} + B) = A \cdot B
\end{aligned}$$

$$\begin{aligned}
& A + (A \cdot B) = (A + A) \cdot (A + B) = \\
& A \cdot (A + B) = A \cdot A + A \cdot B = \\
& A + A \cdot B = A \cdot (1 + B) + A \cdot B = \\
& A + A \cdot B + A \cdot B = A + A(B + B) = \\
& A + A \cdot 1 = A + A = A
\end{aligned}
\quad
\begin{aligned}
& A + (\overline{A} B) = A(1 + B) + \overline{A} B = \\
& A + AB + \overline{A} B = \\
& A + B(A + \overline{A}) = A + B
\end{aligned}$$

Teorema di: **DE MORGAN**

$$\begin{aligned}
10) \quad & \overline{(A + B)} = \overline{A} \cdot \overline{B} \\
& \overline{A \cdot B} = \overline{A} + \overline{B}
\end{aligned}$$

FUNZIONI LOGICHE

- si rappresentano con **tabelle di verità**

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

A	B	$A + B$	$\bar{A}B$	$A + \bar{A}B$
0	0	0	0	0
0	1	1	1	1
1	0	1	0	1
1	1	1	0	1

$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

A	B	C	$B \cdot C$	$A + (B \cdot C)$	$A + B$	$A + C$	$(A + B) \cdot (A + C)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Le funzioni logiche si possono: **semplificare**

Definiamo:

B	A	minterm	maxterm
0	0	$\bar{A} \cdot \bar{B}$	$A + B$
0	1	$\bar{A} \cdot B$	$A + \bar{B}$
1	0	$A \cdot \bar{B}$	$\bar{A} + B$
1	1	$A \cdot B$	$\bar{A} + \bar{B}$

} 2^n

$$\text{minterm} = \overline{\text{maxterm}}$$

$$\text{maxterm} = \overline{\text{minterm}}$$

$$F = \sum_{i=0}^{2^n-1} u_i P_i$$

Esempi di semplificazione:

$$F_1 = \bar{a}b + a\bar{c} + \bar{a}c + a\bar{b} = \bar{a}b + a\bar{c} + \bar{b}c$$

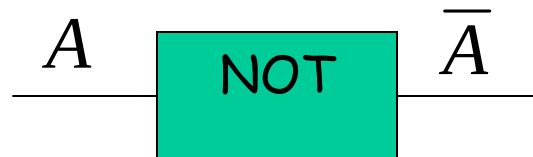
$$F_2 = \bar{a}\bar{b}c + \bar{a}b\bar{c} + \bar{a}bc + a\bar{b}c = \bar{a}\bar{c} + \bar{a}b + a\bar{b}c$$

$$\begin{aligned} F_1 &= \bar{a}b + a\bar{c} + \bar{a}c(b + \bar{b}) + a\bar{b}(c + \bar{c}) = \\ &\bar{a}b + a\bar{c} + \bar{a}cb + \bar{a}c\bar{b} + a\bar{b}c + a\bar{b}\bar{c} = \\ &\bar{a}b(1 + c) + a\bar{c}(1 + \bar{b}) + \bar{b}c(a + \bar{a}) = \end{aligned}$$

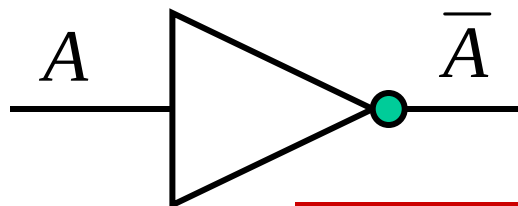
$$\begin{aligned} F_2 &= \bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}bc + a\bar{b}c = \\ &\bar{a}\bar{c}(b + \bar{b}) + \bar{a}bc + a\bar{b}c = \\ &\bar{a}\bar{c}(1 + b) + \bar{a}bc + a\bar{b}c = \\ &\bar{a}\bar{c} + \bar{a}\bar{c}b + \bar{a}bc + a\bar{b}c = \\ &\bar{a}\bar{c} + \bar{a}b(c + \bar{c}) + a\bar{b}c \end{aligned}$$

Le operazioni ed i simboli in elettronica (digitale)

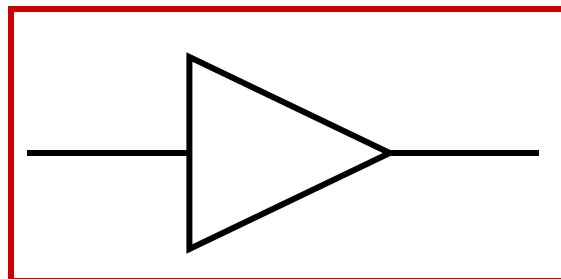
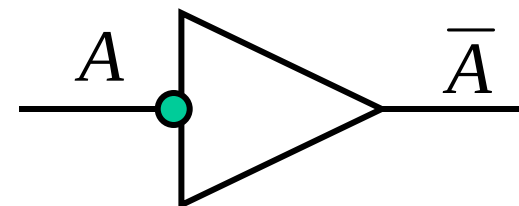
1) NOT



A	X
L	H
H	L



oppure

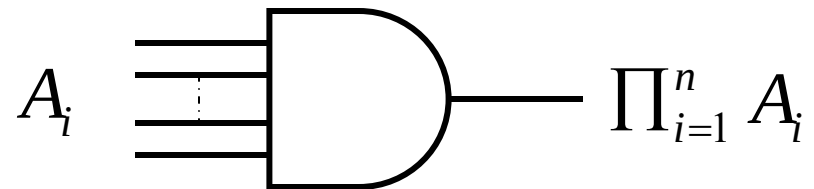
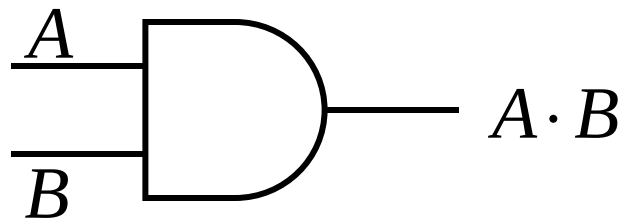


Si chiama BUFFER

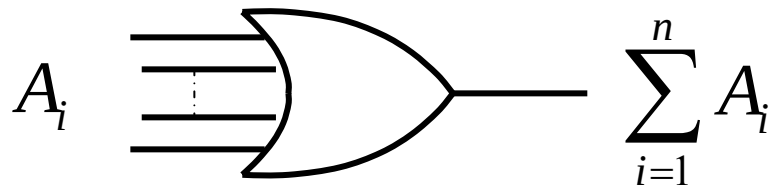
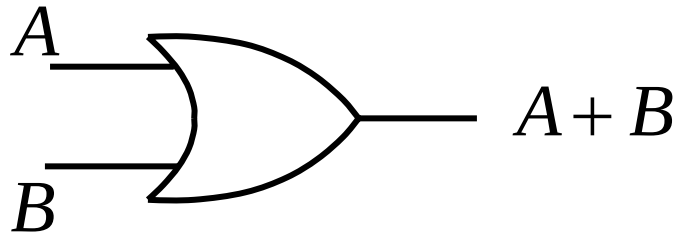
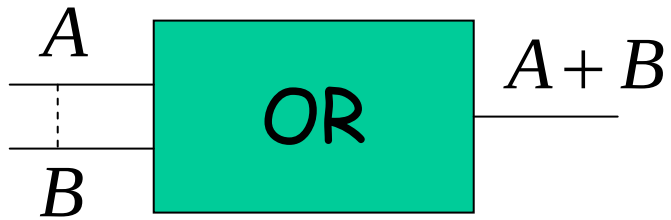
2) AND



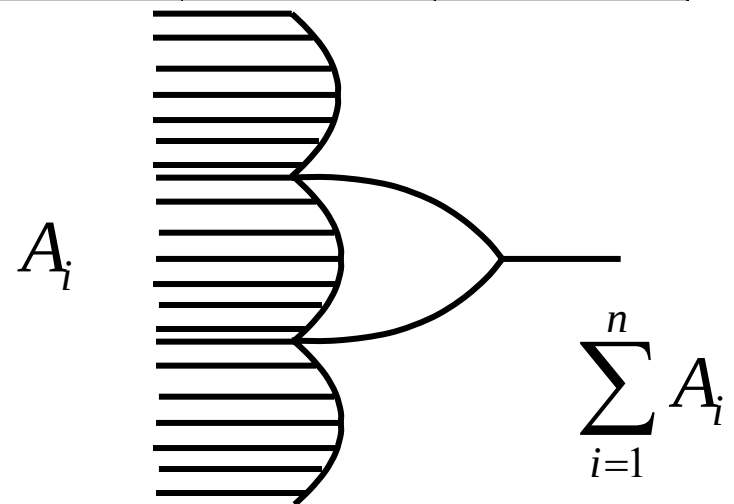
B	A	$A \cdot B$
L	L	L
L	H	L
H	L	L
H	H	H



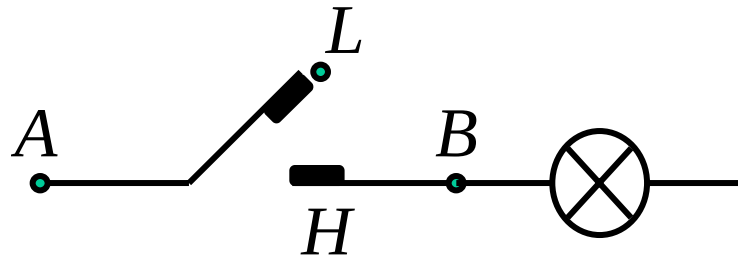
3) OR



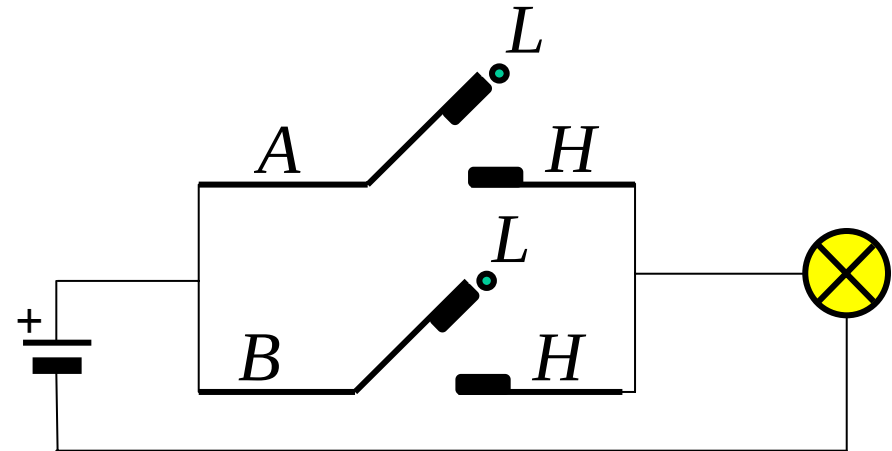
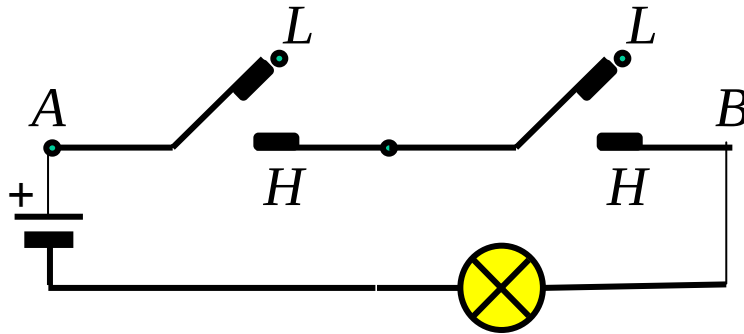
B	A	$A + B$
L	L	L
L	H	H
H	L	H
H	H	H



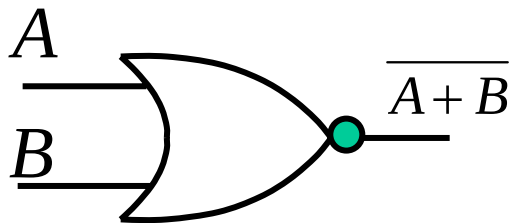
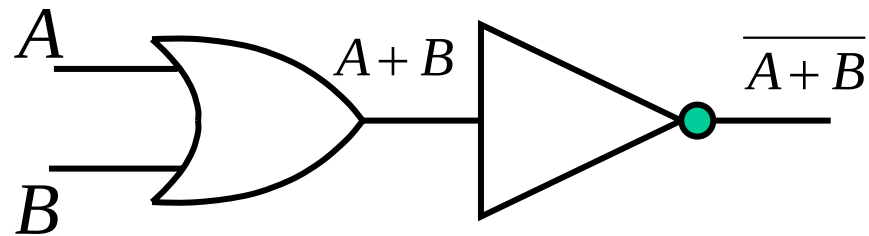
Come li possiamo interpretare?...la preistoria



L' interruttore è un dispositivo a due posizioni (L,H), una delle quali determina la chiusura del contatto elettrico fra i punti A e B mentre l'altra lascia sconnessi i due punti.

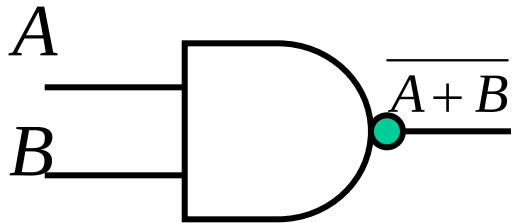
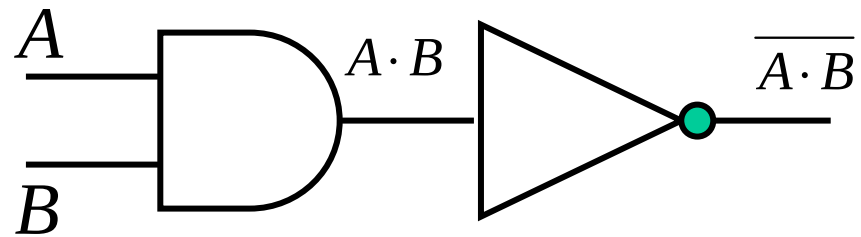


4) NOR



B	A	$\overline{A + B}$
L	L	H
L	H	L
H	L	L
H	H	L

4) NAND



B	A	$\overline{A \cdot B}$
L	L	H
L	H	H
H	L	H
H	H	L

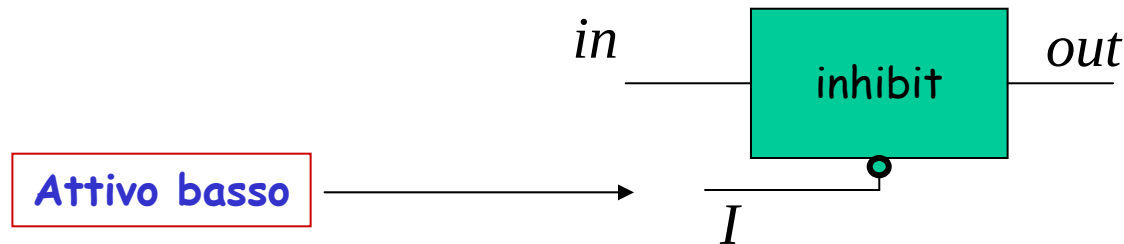
... Ma esistono altri circuiti "non fondamentali"

Enable gate (strobe):

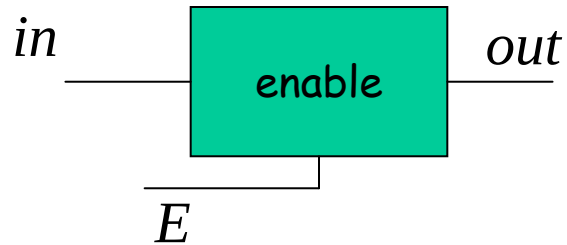
Inhibit gate:



Un comando attivo svolge la funzione relativa al nome del blocco!!

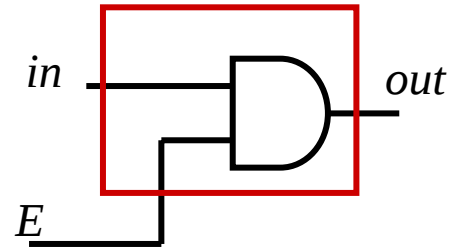


Enable gate (strobe):

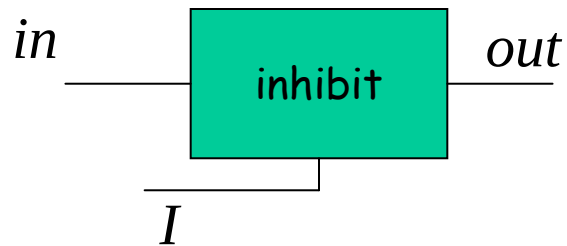


E	In	Out
L	L	L
L	H	L
H	L	L
H	H	H

$$Out = E \cdot In$$

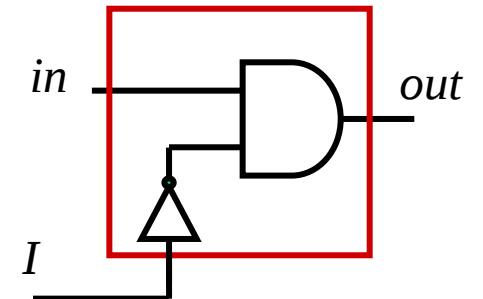


Inhibit gate:

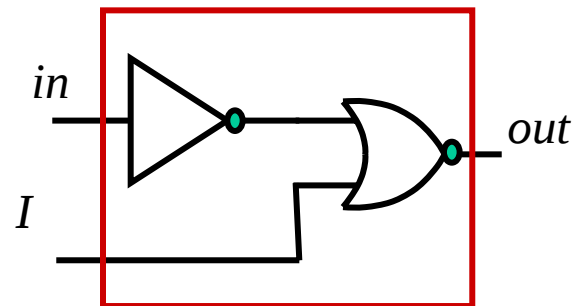


I	In	Out
L	L	L
L	H	H
H	L	L
H	H	L

$$Out = \bar{I} \cdot In$$

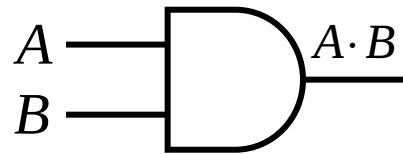


.....e con De Morgan:



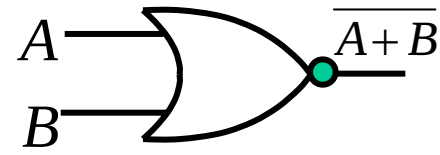
Altre funzioni di due variabili:

AND



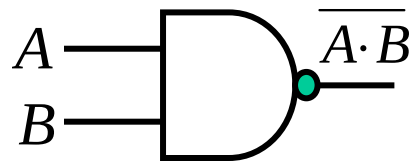
B	A	AB
0	0	0
0	1	0
1	0	0
1	1	1

NOR



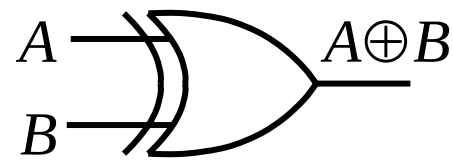
B	A	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

NAND



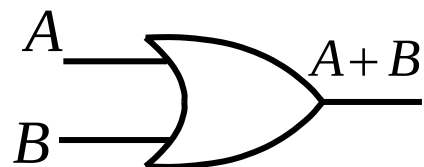
B	A	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

EXOR



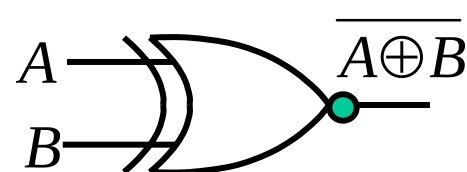
B	A	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

OR



B	A	A+B
0	0	0
0	1	1
1	0	1
1	1	1

EXNOR



B	A	$\overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

Forma "normale" di una funzione "combinatoria"

Somme di prodotti

$$X = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD$$

$$\overline{X} = \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + ABCD$$

A	B	C	D	X	$\overline{X}$	minterm
0	0	0	0	1	0	$\overline{A}\overline{B}\overline{C}\overline{D}$
0	0	0	1	0	1	$\overline{A}\overline{B}\overline{C}D$
0	0	1	0	0	1	$\overline{A}\overline{B}C\overline{D}$
0	0	1	1	1	0	$\overline{A}\overline{B}CD$
0	1	0	0	1	0	$\overline{A}B\overline{C}\overline{D}$
0	1	0	1	0	1	$\overline{A}B\overline{C}D$
0	1	1	0	0	1	$\overline{A}BC\overline{D}$
0	1	1	1	0	1	$\overline{A}BCD$
1	0	0	0	1	0	$A\overline{B}\overline{C}\overline{D}$
1	0	0	1	1	0	$A\overline{B}\overline{C}D$
1	0	1	0	1	0	$A\overline{B}C\overline{D}$
1	0	1	1	0	1	$A\overline{B}CD$
1	1	0	0	0	1	$AB\overline{C}\overline{D}$
1	1	0	1	0	1	$AB\overline{C}D$
1	1	1	0	0	1	$ABC\overline{D}$
1	1	1	1	1	0	$ABCD$

$$X = (A + B + C + \overline{D}) \cdot (A + B + \overline{C} + D) \cdot (A + \overline{B} + C + \overline{D}) \cdot (A + \overline{B} + \overline{C} + D) \cdot (\overline{A} + B + \overline{C} + \overline{D}) \cdot (\overline{A} + B + C + D) \cdot (\overline{A} + \overline{B} + C + \overline{D}) \cdot (\overline{A} + \overline{B} + \overline{C} + D)$$

Prodotti di Somme (si Prendono gli Zer!!!!)

Dimostriamo: Teorema di DE MORGAN (dalle tabelle di verità)

Partiamo dalla **AND**:

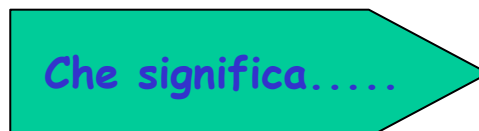
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1



$\bar{A}$	$\bar{B}$	$\overline{A \cdot B}$
1	1	1
1	0	1
0	1	1
0	0	0

Cambiamo i nomi: $\bar{A} = C$ $\bar{B} = D$ $\bar{X} = Y$

C	D	Y
0	0	0
0	1	1
1	0	1
1	1	1



$$Y = C + D$$

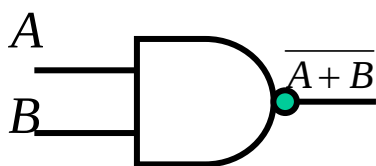
$$\overline{\bar{A} \cdot \bar{B}} = A + B \qquad \overline{\bar{A} + \bar{B}} = A \cdot B$$

N.B.

Se uno schema logico (elettronico) realizza una certa funzione, per ottenerne il complemento basta scambiare le AND con le OR e complementare le variabili di ingresso

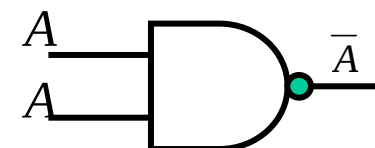
Porte universali.....cosa vuol dire?

1. NAND

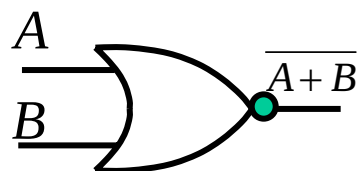


B	A	$\overline{A \cdot B}$
L	L	H
L	H	H
H	L	H
H	H	L

Cosa sono?

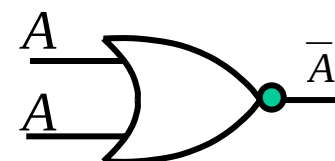


2. NOR



B	A	$\overline{A + B}$
L	L	H
L	H	L
H	L	L
H	H	L

Cosa sono?



I
N
V
E
R
T
I
O
R
H

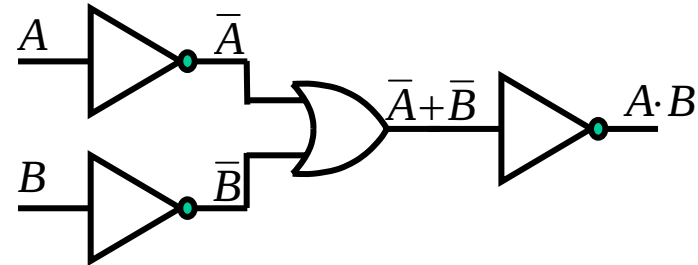
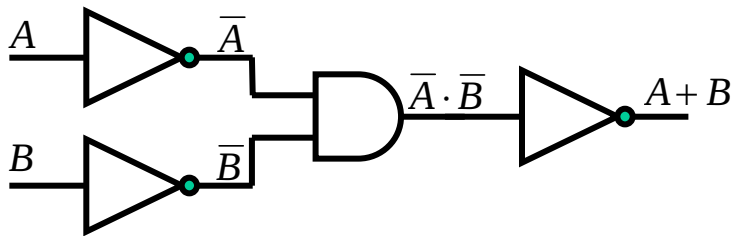
Porte universali

Dal teorema di De Morgan:

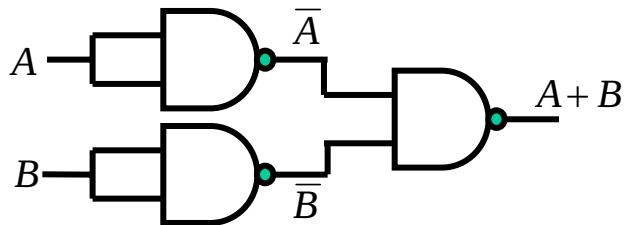
$$\overline{\overline{A} \cdot \overline{B}} = A + B$$

$$\overline{\overline{A} + \overline{B}} = A \cdot B$$

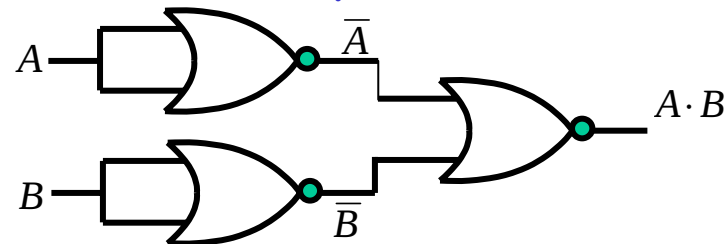
E' sufficiente realizzare il circuito corrispondente al primo membro e verificare che la sua tavola della verità sia uguale a quella del secondo membro.



Or con solo porte NAND



AND con solo porte NOR

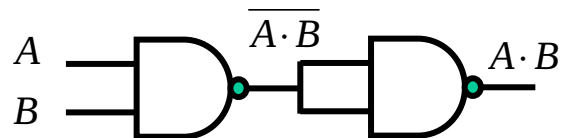
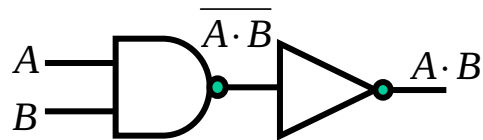


Porte universali

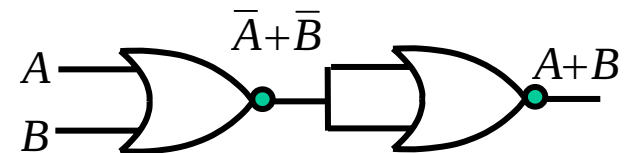
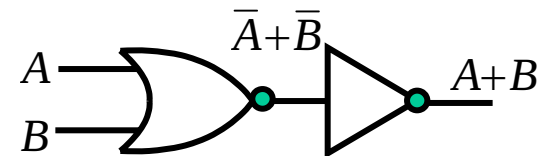
And con solo porte NAND

Or con solo porte NOR

$$\overline{\overline{A \cdot B}} = A \cdot B$$



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



**Lezioni di:
Laboratorio di elettronica digitale**

Anno Accademico 2004-2005

Parte Seconda

Esperienze di Laboratorio

Scopo del Laboratorio di Elettronica Digitale

Circuiti: Analogici, Logici

sottosistemi a componenti: Discreti, Integrati

Realizzazioni di prototipi: uso delle breadboard

Circuiti Integrati: chip monolitici

Due grossi gruppi:

- gruppo bipolare: cariche positive e negative
- gruppo unipolare: cariche di una sola polarità

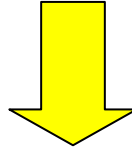
I gruppi si dividono in famiglie

Le famiglie sono caratterizzate da caratteristiche salienti

All' interno delle famiglie i CI sono tutti compatibili tra loro

- Stessi livelli
- Stesse alimentazioni
- Potenze compatibili

Tra famiglie diverse i CI sono (in generale) incompatibili tra loro



Circuiti di interfaccia

I principali parametri che caratterizzano le famiglie:

1. Ritardo di propagazione:maggiore nei circuiti unipolari
2. Dissipazione di potenza:inferiore negli unipolari
3. Capacità di pilotaggio (fan-out):maggiore negli unipolari
4. Immunità al rumore:migliore negli unipolari
5. Capacità di una porta (fan-in): equivalente
6. Densità di integrazione:maggiore negli unipolari

La scelta va fatta in base alle caratteristiche/necessità di progetto

Famiglie Bipolari:

1. RTL:obsoleta
2. DTL:obsoleta
3. HTL:
4. TTL standard:
5. TTL a bassa dissipazione:
6. TTL high speed:
7. TT Schottky:
8. ECL:
9. I²L:

Famiglie unipolari:

1. P-MOS H.V.:
2. P-MOS L.V.:
3. N-MOS:
4. C-MOS:

Scale di Integrazione:

1. **S**(mall)**S**(cale)**I**(ntegration):
12 porte (50 transistor equivalenti)
2. **M**(edium)**S**(cale)**I**(tegration):
12-100 porte (50-500 trs equivalenti)
3. **L**(arge)**S**(cale)**I**(ntegration):
100-1000 porte (500-4000 trs equivalenti)
4. **V**(ery)**L**(arge)**S**(cale)**I**(ntegration):
:
>1000 porte ($\geq 10^7$ trs eq. per il PENTIUM INTEL nel 2002)

TI LOGIC DEVICE NOMENCLATURE

Are you confused by the device names and numbers of logic products? Do you need to know if a part has bus hold or series damping resistors? What kind of package the part is available in? The table below takes the mystery out of what all the letters and numbers in TI standard catalog logic devices represent. Take your favorite TI logic part number and find out what it can do.

SN	74	LVC	H	16	2	244	A	DGG	R
1	2	3	4	5	6	7	8	9	10

1. Standard Prefix

- Example: SNJ - Conforms to MIL-PRF-38535 (QML)

2. Temperature Range

- 54 - Military
- 74 - Comercial

3. Family

4. Special Features

- Blank = No Special Features
- C - Configurable Vcc (LVCC)
- D - [Level-Shifting Diode](#) (CBTD)
- H - [Bus Hold](#) (ALVCH)
- K - Undershoot-Protection Circuitry (CBTK)
- R - [Damping Resistor on Inputs/Outputs](#) (LVCR)
- S - Schottky Clamping Diode (CBTS)
- Z - Power-Up 3-State (LVCZ)

5. Bit Width

- Blank = Gates, MSI, and Octals
- 1G - Single Gate
- 8 - Octal IEEE 1149.1 ([JTAG](#))
- 16 - [Widebus™](#) (16, 18, and 20 bit)
- 18 - Widebus IEEE 1149.1 ([JTAG](#))
- 32 - [Widebus+™](#) (32 and 36 bit)

6. Options

- Blank = No Options
- 2 - [Series-Damping Resistor on Outputs](#)
- 4 - Level Shifter
- 25 - 25-ohm Line Driver

7. Function

- 244 - Noninverting Buffer/Driver
- 374 - D-Type Flip-Flop
- 573 - D-Type Transparent Latch
- 640 - Inverting Transceiver

8. Device Revision

- Blank = No Revision
- Letter Designator A-Z

9. Packages

- D, DW - Small-Outline Integrated Circuit (SOIC)
- DB, DL - Shrink Small-Outline Package (SSOP)
- DBB, DGV - Thin Very Small-Outline Package (TVSOP)
- DBQ - Quarter-Size Outline Package (QSOP)
- DBV, DCK - Small-Outline Transistor Package (SOT)
- DGG, PW - Thin Shrink Small-Outline Package (TSSOP)
- FK - Leadless Ceramic Chip Carrier (LCCC)
- FN - Plastic Leaded Chip Carrier (PLCC)
- GB - Ceramic Pin Grid Array (CPGA)
- GKE, GKF - MicroStar BGA™ Low-Profile Fine-Pitch Ball Grid Array (LFBGA)
- GQL, GQN - MicroStar Junior BGA Very-Thin-Profile Fine-

Pitch Ball Grid Array (VFBGA)

- HFP, HS, HT, HV - Ceramic Quad Flat Package (CQFP)
- J, JT - Ceramic Dual-In-Line Package (CDIP)
- N, NP, NT - Plastic Dual-In-Line Package (PDIP)
- NS, PS - Small Outline Package (SOP)
- PAG, PAH, PCA, PCB, PM, PN, PZ - Thin Quad Flat Package (TQFP)
- PH, PQ, RC - Quad Flat Package (QFP)
- W, WA, WD - Ceramic Flat Package (CFP)

10. Tape and Reel

All new or changed devices in the **DB** and **PW** package types include the **R** designation for reeled product. Existing products designated as **LE** presently maintain that designation, but will be converted to **R** in the future.

Nomenclature Examples:

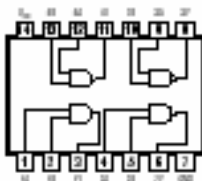
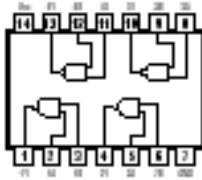
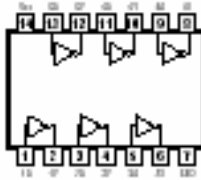
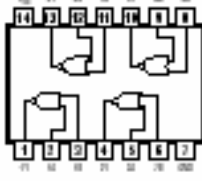
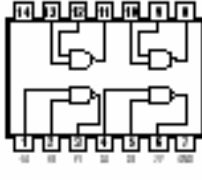
- For an Existing Device - SN74LVTxxxDB**LE**
- For a New or Changed Device - SN74LVTxxxADB**R**
- **LE** - Left Embossed (valid for DB and PW packages only)
- **R** - Standard (valid for all surface-mount packages except existing DB and PW devices)

Family Comparison

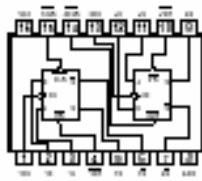
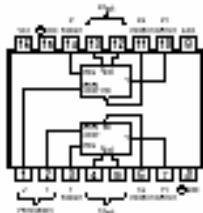
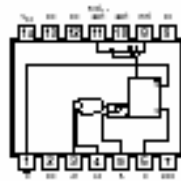
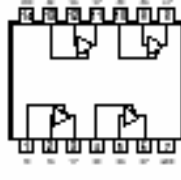
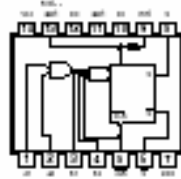
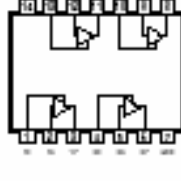
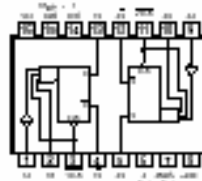
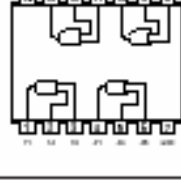
The table below compares some typical characteristics of several popular logic families available in the market today. The following sections provide brief explanations of the various parameters.

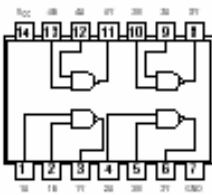
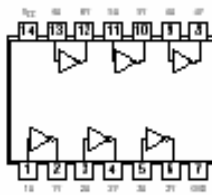
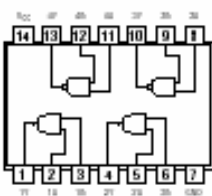
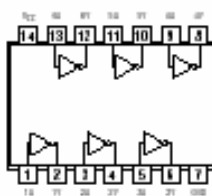
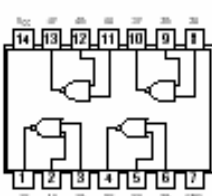
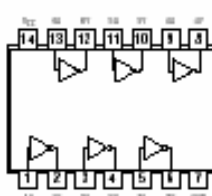
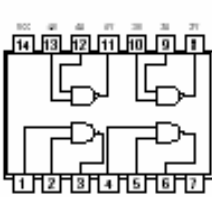
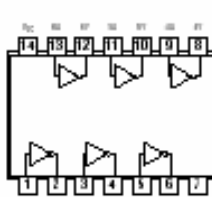
Typical Commercial Parameter (0°C to +70°C)	Logic Families												
	TTL				CMOS					ECL			
	LS	ALS	ABT	FAST	MG	HC	FACT	LVC	LCX	10H	100K	ECL in PS(3)	E-Lite
Speed Gate Prop Delay (ns)	9	7	2.7	3	65	8	5	3.3	3.5	1	0.75	0.33	0.22
Flip-Flop Toggle Rate (MHz)	33	45	200	125	4	45	160	200	200	330	400	1,000	2800
Output Edge Rate (ns)	6	3	3	2	50	4	2	3.7	3.6	1	0.7	0.5	0.25
Power Consumption Per Gate (mW)													
Quiescent	5	1.2	0.005	12.5	0.0006	0.003	0.0001	0.003	1E-04	25	50	25	73
Operating (1 MHz)	5	1.2	1.0	12.5	0.04	0.6	0.6	0.8	0.3	25	50	25	73
Supply Voltage (V)	+4.5 to +5.5	+4.5 to +5.5	+4.5 to +5.5	+4.5 to +5.5	+3 to +18	+2 to +6	+1.2 to +3.6	+2 to +3.6	+2 to +6	-4.5 to -5.5	-4.2 to -4.8	-4.2 to -5.5	-4.2 to -5.5
Output Drive (mA)	8	8	32/64	20	1	4	24	24	24	50 ohm load	50 ohm load	50 ohm load	50 ohm load

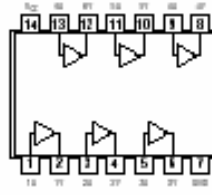
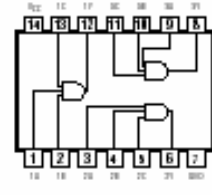
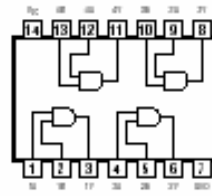
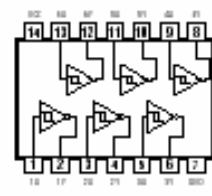
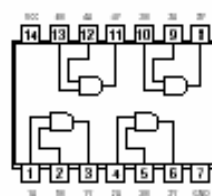
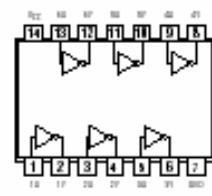
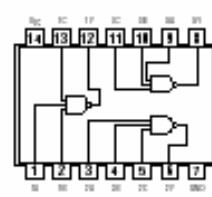
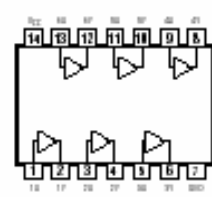
Pin Assignments

00 QUADRUPLE 2-INPUT POSITIVE-NAND GATES positive logic: $Y = A \cdot B$  See page 138	04 HEX INVERTERS positive logic: $Y = \bar{X}$  See page 143
01 QUADRUPLE 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS positive logic: $Y = A \cdot B$  See page 140	U04 HEX INVERTERS positive logic: $Y = \bar{X}$  See page 143
02 QUADRUPLE 2-INPUT POSITIVE-NOR GATES positive logic: $Y = A + B$  See page 141	05 HEX INVERTERS WITH OPEN-COLLECTOR OUTPUTS positive logic: $Y = \bar{X}$  See page 143
03 QUADRUPLE 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS positive logic: $Y = A \cdot B$  See page 142	06 HEX INVERTER BUFFERS/DRIVERS WITH OPEN-COLLECTOR HIGH-VOLTAGE OUTPUTS positive logic: $Y = X$  See page 143

Pin Assignments

112 DUAL J-K NEGATIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR  See page 138	124 DUAL VOLTAGE-CONTROLLED OSCILLATORS WITH ENABLE INPUTS  See page 138
121 MONOSTABLE MULTIVIBRATOR  See page 138	125 QUADRUPLE BUS BUFFER GATES WITH THREE-STATE OUTPUTS positive logic: $Y = A$  See page 138
122 RETRIGGERABLE MONOSTABLE MULTIVIBRATORS WITH CLEAR  See page 138	126 QUADRUPLE BUS BUFFER GATES WITH THREE-STATE OUTPUTS positive logic: $Y = A$  See page 138
123 DUAL RETRIGGERABLE MONOSTABLE MULTIVIBRATORS WITH CLEAR  See page 138	128 SN54128L, 75-Q LINE DRIVER SN74128L, 58-Q LINE DRIVER positive logic: $Y = A + B$  See page 138

00 QUADRUPL 2-INPUT POSITIVE-NAND GATES positive logic: $Y = \overline{A \cdot B}$  See page 139	04 HEX INVERTERS positive logic: $Y = \overline{A}$  See page 140
01 QUADRUPL 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS positive logic: $Y = \overline{A \cdot B}$  See page 140	U04 HEX INVERTERS positive logic: $Y = \overline{A}$  See page 146
02 QUADRUPL 2-INPUT POSITIVE-NOR GATES positive logic: $Y = \overline{A + B}$  See page 141	05 HEX INVERTERS WITH OPEN-COLLECTOR OUTPUTS positive logic: $Y = \overline{A}$  See page 146
03 QUADRUPL 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS positive logic: $Y = \overline{A \cdot B}$  See page 142	06 HEX INVERTER BUFFERS/DRIVERS WITH OPEN-COLLECTOR HIGH-VOLTAGE OUTPUTS positive logic: $Y = \overline{A}$  See page 145

07 HEX BUFFERS/DRIVERS WITH OPEN-COLLECTOR HIGH-VOLTAGE OUTPUTS positive logic: $Y = \overline{A}$  See page 145	11 TRIPLE 3-INPUT POSITIVE-AND GATES positive logic: $Y = A \cdot B \cdot C$  See page 149
08 QUADRUPL 2-INPUT POSITIVE-AND GATES positive logic: $Y = A \cdot B$  See page 145	14 HEX SCHMITT-TRIGGER INVERTERS positive logic: $Y = \overline{A}$  See page 150
09 QUADRUPL 2-INPUT POSITIVE-AND GATES WITH OPEN-COLLECTOR OUTPUTS positive logic: $Y = A \cdot B$  See page 147	16 HEX INVERTER BUFFERS/DRIVERS WITH OPEN-COLLECTOR HIGH-VOLTAGE OUTPUTS positive logic: $Y = \overline{A}$  See page 151
10 TRIPLE 3-INPUT POSITIVE-NAND GATES positive logic: $Y = \overline{A \cdot B \cdot C}$  See page 148	17 HEX BUFFERS/DRIVERS WITH OPEN-COLLECTOR HIGH-VOLTAGE OUTPUTS positive logic: $Y = \overline{A}$  See page 151

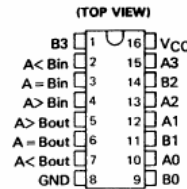
SN5485, SN54LS85, SN54S85
SN7485, SN74LS85, SN74S85
4-BIT MAGNITUDE COMPARATORS
SDLS123 - MARCH 1974 - REVISED MARCH 1988

TYPE	TYPICAL POWER DISSIPATION	TYPICAL DELAY (4-BIT WORDS)
'85	275 mW	23 ns
LS85	52 mW	24 ns
S85	365 mW	11 ns

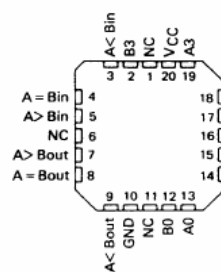
description

These four-bit magnitude comparators perform comparison of straight binary and straight BCD (8-4-2-1) codes. Three fully decoded decisions about two 4-bit words (A, B) are made and are externally available at three outputs. These devices are fully expandable to any number of bits without external gates. Words of greater length may be compared by connecting comparators in cascade. The $A > B$, $A < B$, and $A = B$ outputs of a stage handling less-significant bits are connected to the corresponding $A > B$, $A < B$, and $A = B$ inputs of the next stage handling more-significant bits. The stage handling the least-significant bits must have a high-level voltage applied to the $A = B$ input. The cascading paths of the '85, LS85, and S85 are implemented with only a two-gate-level delay to reduce overall comparison times for long words. An alternate method of cascading which further reduces the comparison time is shown in the typical application data.

SN5485, SN54LS85, SN54S85 . . . J OR W PACKAGE
SN7485 . . . N PACKAGE
SN74LS85, SN74S85 . . . D OR N PACKAGE



SN54LS85, SN54S85 . . . FK PACKAGE
(TOP VIEW)



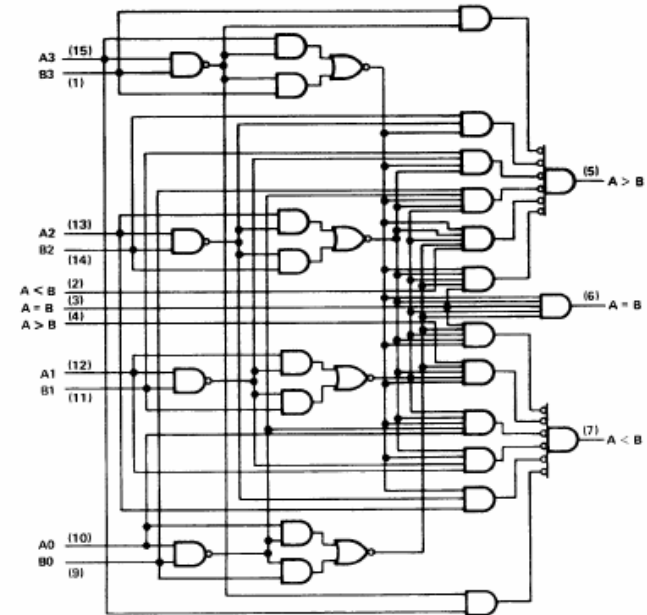
NC - No internal connection

FUNCTION TABLE

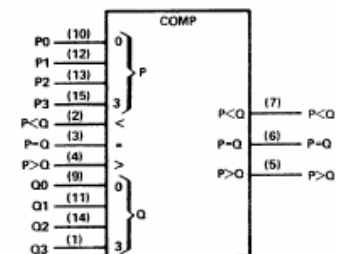
COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A3, B3	A2, B2	A1, B1	A0, B0	A > B	A < B	A = B	A > B	A < B	A = B
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 = B3	A2 < B2	X	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A2 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

SN5485, SN54LS85, SN54S85
SN7485, SN74LS85, SN74S85
4-BIT MAGNITUDE COMPARATORS
SDLS123 - MARCH 1974 - REVISED MARCH 1988

logic diagrams (positive logic)



logic symbol†



†This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12. Pin numbers shown are for D, J, N, and W packages.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

**TEXAS
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 1988, Texas Instruments Incorporated

1

2

**TEXAS
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

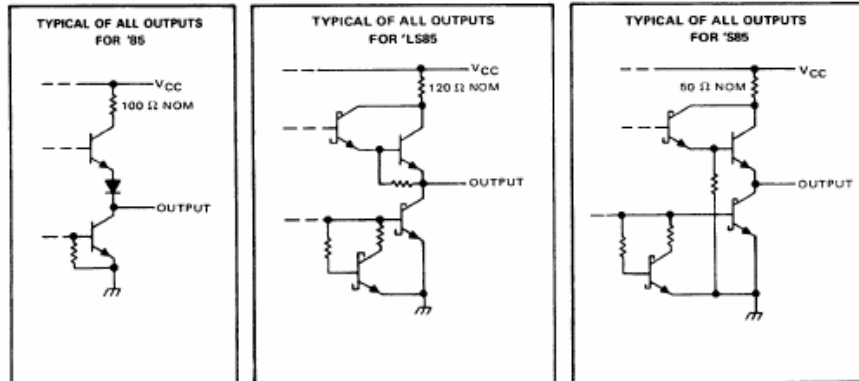
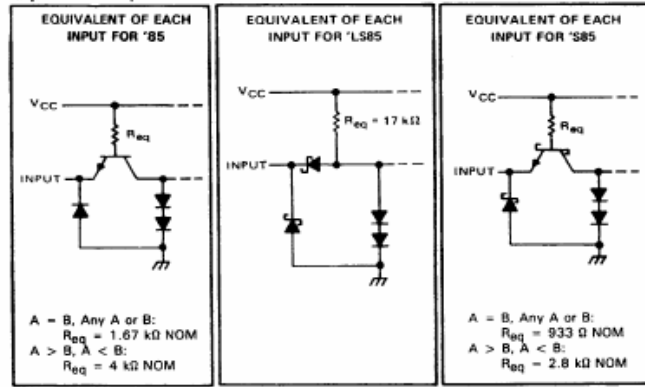
A.A. 2006-07
3° trimestre

Prof. M.Savrié savrie@fe.infn.it
<http://www.fe.infn.it/~savrie/>

40

SN5485, SN54LS85, SN54S85
SN7485, SN74LS85, SN74S85
4-BIT MAGNITUDE COMPARATORS
SDLS123 - MARCH 1974 - REVISED MARCH 1985

schematics of inputs and outputs



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

	SN54 ¹ SN54S ¹	SN54LS ¹	SN74 ¹ SN74S ¹	SN74LS ¹	UNIT
Supply voltage, V_{CC} (see Note 1)	7	7	7	7	V
Input voltage	5.5	7	5.5	7	V
Intermittent voltage (see Note 2)	5.5		5.5		V
Operating free-air temperature range	-55 to 125		-0 to 70		°C
Storage temperature range	-65 to 150		-65 to 150		°C

NOTES: 1. Voltage values, except intermittent voltage, are with respect to network ground terminal.
2. This is the voltage between two emitters of a multiple-emitter input transistor. This rating applies to each A input in conjunction with its respective B input of the '85 and 'S85.

SN5485, SN54LS85, SN54S85
SN7485, SN74LS85, SN74S85
4-BIT MAGNITUDE COMPARATORS
SDLS123 - MARCH 1974 - REVISED MARCH 1985

recommended operating conditions

	SN5485			SN7485			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			-400			-400	μA
Low-level output current, I_{OL}			16			16	μA
Operating free-air temperature, T_A	-55	125	0	70			°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS ¹	MIN	TYP ²	MAX	UNIT
V_{IH} High-level input voltage			2		V
V_{IL} Low-level input voltage				0.8	V
V_{IK} Input clamp voltage	$V_{CC} = \text{MIN.}$, $I_I = -12 \text{ mA}$			-1.5	V
V_{OH} High-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IL} = 0.8 \text{ V}$, $I_{OH} = -400 \mu\text{A}$	2.4	3.4		V
V_{OL} Low-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $I_{OL} = 16 \text{ mA}$	0.2	0.4		V
I_I Input current at maximum input voltage	$V_{CC} = \text{MAX.}$, $V_I = 5.5 \text{ V}$			1	mA
I_{IH} High-level input current	A < B, A > B inputs all other inputs $V_{CC} = \text{MAX.}$, $V_I = 2.4 \text{ V}$			40	μA
I_{IL} Low-level input current	A < B, A > B inputs all other inputs $V_{CC} = \text{MAX.}$, $V_I = 0.4 \text{ V}$			-1.6	mA
I_{OS} Short-circuit output current ³	$V_{CC} = \text{MAX.}$, $V_O = 0$			-55	mA
I_{CC} Supply current	$V_{CC} = \text{MAX.}$ See Note 4			55	mA

¹ For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

² All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

³ Not more than one output should be shorted at a time.

NOTE 4: I_{CC} is measured with outputs open, A = B grounded, and all other inputs at 4.5 V.

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$

PARAMETER ¹	FROM INPUT	TO OUTPUT	NUMBER OF GATE LEVELS	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Any A or B data input	A < B, A > B	1	$C_L = 15 \text{ pF}$, $R_L = 400 \Omega$, See Note 5		7		ns
			2			12		
			3			17	26	
		A = B	4			23	35	ns
t_{PHL}	Any A or B data input	A < B, A > B	1			11		
			2			15		
			3			20	30	ns
		A = B	4			20	30	
t_{PLH}	A < B or A = B	A > B	1			7	11	ns
t_{PHL}	A < B or A = B	A > B	1			11	17	
t_{PLH}	A = B	A > B	2			13	20	ns
t_{PHL}	A = B	A > B	2			11	17	

¹ t_{PLH} = propagation delay time, low-to-high-level output

t_{PHL} = propagation delay time, high-to-low-level output

NOTE 5: Load circuits and voltage waveforms are shown in Section 1.



POST OFFICE BOX 655033 • DALLAS, TEXAS 75265

3



POST OFFICE BOX 655033 • DALLAS, TEXAS 75265

4

A.A. 2006-07
3° trimestre

Prof. M.Savrié savrie@fe.infn.it
<http://www.fe.infn.it/~savrie/>

41

SN5485, SN54LS85, SN54S85
SN7485, SN74LS85, SN74S85
4-BIT MAGNITUDE COMPARATORS
SDLS 123 - MARCH 1974 - REVISED MARCH 1985

recommended operating conditions

	SN54LS85			SN74LS85			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			-400			-400	μ A
Low-level output current, I_{OL}			4			8	mA
Operating free-air temperature, T_A	-55		125	0		70	$^{\circ}$ C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS [†]	SN54LS85			SN74LS85			UNIT
			MIN	TYP [‡]	MAX	MIN	TYP [‡]	MAX	
V_{IH}	High-level input voltage		2			2			V
V_{IL}	Low-level input voltage			0.7			0.7		V
V_{IK}	Input clamp voltage	$V_{CC} = \text{MIN.}$, $I_I = -18 \text{ mA}$		-1.5			-1.5		V
V_{OH}	High-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $V_{IL} = V_{IL \text{ max.}}$, $I_{OH} = -400 \mu\text{A}$	2.5	3.4		2.7	3.4		V
V_{OL}	Low-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $V_{IL} = V_{IL \text{ max.}}$, $I_{OL} = 8 \text{ mA}$		0.25	0.4		0.25	0.4	V
I_I	Input current at maximum input voltage	$V_{CC} = \text{MAX.}$, $V_I = 7 \text{ V}$		0.1			0.1		mA
I_{IH}	High-level input current	$V_{CC} = \text{MAX.}$, $V_I = 2.7 \text{ V}$		20			60		μ A
I_{IL}	Low-level input current	$V_{CC} = \text{MAX.}$, $V_I = 0.4 \text{ V}$		-0.4			-1.2		mA
I_{OS}	Short-circuit output current [§]	$V_{CC} = \text{MAX.}$	-20	-100	-20	-100			mA
I_{CC}	Supply current	$V_{CC} = \text{MAX.}$, See Note 4	10.4	20		10.4	20		mA

[†]For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

[‡]All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^{\circ}\text{C}$.

[§]Not more than one output should be shorted at a time, and duration of the short circuit should not exceed one second.

NOTE 4: I_{CC} is measured with outputs open, A = B grounded, and all other inputs at 4.5 V.

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^{\circ}\text{C}$

PARAMETER [§]	FROM INPUT	TO OUTPUT	NUMBER OF GATE LEVELS	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Any A or B data input	A < B, A > B	1	$C_L = 15 \text{ pF}$, $R_L = 2 \text{ k}\Omega$, See Note 5		14		ns
			2			19		
			3			24	36	
			4			27	45	
t_{PHL}	Any A or B data input	A < B, A > B	1			11		ns
			2			15		
			3			20	30	
			4			23	45	
t_{PLH}	A < B or A = B	A > B	1			14	22	ns
t_{PHL}	A < B or A = B	A > B	1			11	17	ns
t_{PLH}	A = B	A < B	2			13	20	ns
t_{PHL}	A = B	A < B	2			13	26	ns
t_{PLH}	A > B or A = B	A < B	1			14	22	ns
t_{PHL}	A > B or A = B	A < B	1			11	17	ns

[§] t_{PLH} = propagation delay time, low-to-high-level output

t_{PHL} = propagation delay time, high-to-low-level output

NOTE 5: Load circuits and voltage waveforms are shown in Section 1.

SN5485, SN54LS85, SN54S85
SN7485, SN74LS85, SN74S85
4-BIT MAGNITUDE COMPARATORS
SDLS 123 - MARCH 1974 - REVISED MARCH 1985

recommended operating conditions

	SN54S85			SN74S85			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			-1			-1	mA
Low-level output current, I_{OL}			20			20	mA
Operating free-air temperature, T_A	-55		125	0		70	$^{\circ}$ C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS†	MIN	TYP‡	MAX	UNIT	
V _{IH}	High-level input voltage				2		V	
V _{IL}	Low-level input voltage					0.8	V	
V _{IK}	Input clamp voltage		V _{CC} = MIN., I _I = -18 mA			-1.2	V	
V _{OH}	High-level output voltage		V _{CC} = MIN., V _{IH} = 2 V, V _{IL} = 0.8 V, I _{OH} = -1 mA	SN54685	2.5	3.4	V	
				SN74685	2.7	3.4		
V _{OL}	Low-level output voltage		V _{CC} = MIN., V _{IH} = 2 V, V _{IL} = 0.8 V, I _{OL} = 20 mA			0.5	V	
I _I	Input current at maximum input voltage		V _{CC} = MAX., V _I = 5.5 V			1	mA	
I _{IH}	High-level input current	A < B, A > B inputs all other inputs	V _{CC} = MAX., V _I = 2.7 V			50	μA	
						150		
I _{IL}	Low-level input current	A < B, A > B inputs all other inputs	V _{CC} = MAX., V _I = 0.5 V			-2	mA	
						-6		
I _{OS}	Short-circuit output current§		V _{CC} = MAX.			-40	-100	mA
I _{CC}	Supply current		V _{CC} = MAX.			73	115	mA
			V _{CC} = MAX., See Note 4					
			V _{CC} = MAX., T _A = 125° C, See Note 4	SN54S85W			110	

[†]For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

[‡]All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^{\circ}\text{C}$.

[§]Not more than one output should be shorted at a time, and duration of the short circuit should not exceed one second.

NOTE 4: I_{CC} is measured with outputs open, A = B grounded, and all other inputs at 4.5 V.

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^{\circ}\text{C}$

PARAMETER [§]	FROM INPUT	TO OUTPUT	NUMBER OF GATE LEVELS	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Any A or B data input	A < B, A > B	1	$C_L = 15 \text{ pF}$, $R_L = 280 \Omega$, See Note 5		5		ns
			2			7.5		
			3			10.5	16	
			4			12	18	
t_{PHL}	Any A or B data input	A < B, A > B	1			5.5		ns
			2			7		
			3			11	16.5	
			4			11	16.5	
t_{PLH}	A < B or A = B	A > B	1			5	7.5	ns
t_{PHL}	A < B or A = B	A > B	1			5.5	8.5	ns
t_{PLH}	A = B	A < B	2			7	10.5	ns
t_{PHL}	A = B	A < B	2			6	7.5	ns
t_{PLH}	A > B or A = B	A < B	1			5	7.5	ns
t_{PHL}	A > B or A = B	A < B	1			5.5	8.5	ns

[§] t_{PLH} = propagation delay time, low-to-high-level output

t_{PHL} = propagation delay time, high-to-low-level output

NOTE 5: Load circuits and voltage waveforms are shown in Section 1.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

A.A. 2006-U /
3° trimestre

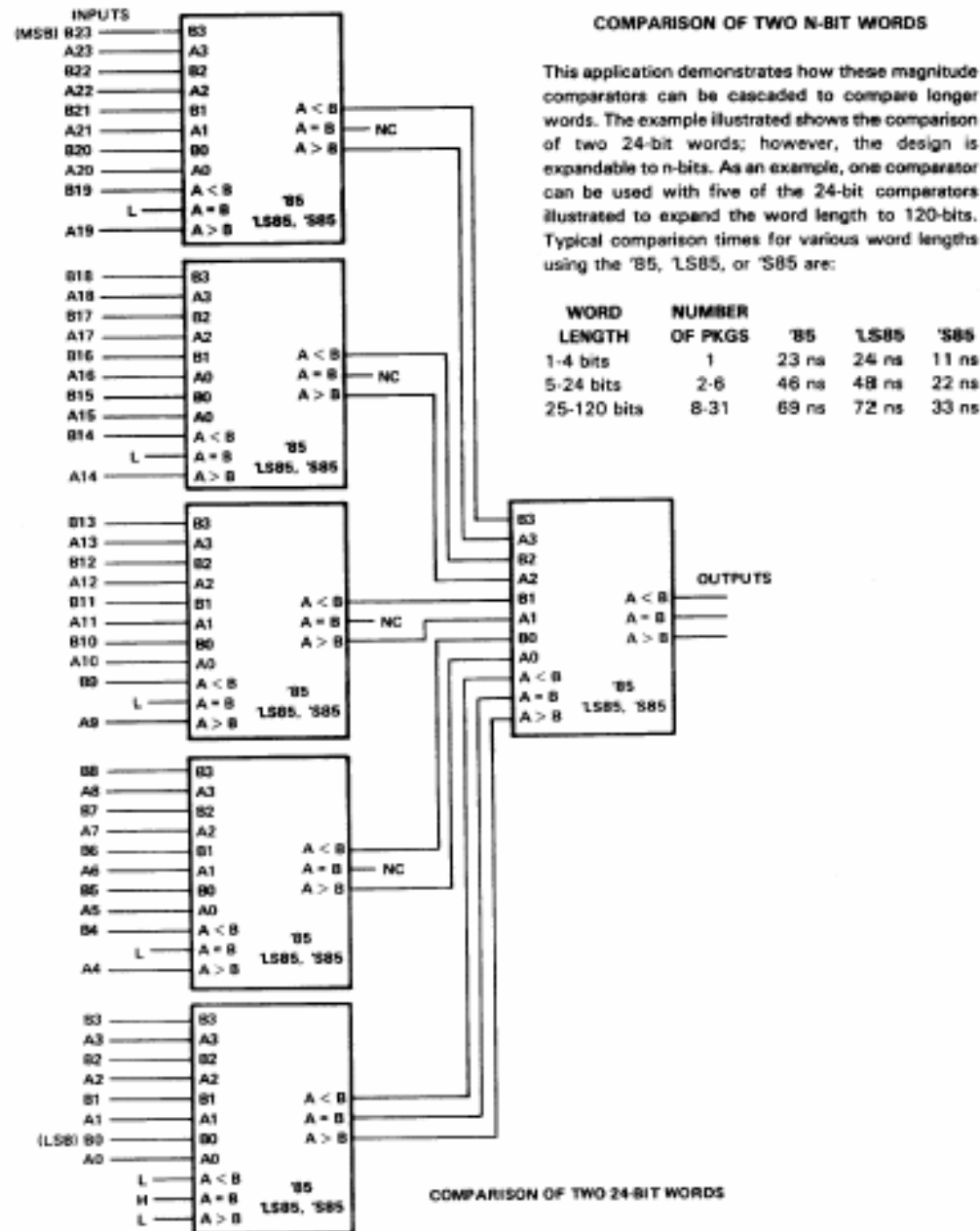
Prof. M. Savrié savrie@fe.infn.it
<http://www.fe.infn.it/~savrie/>



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

4/2

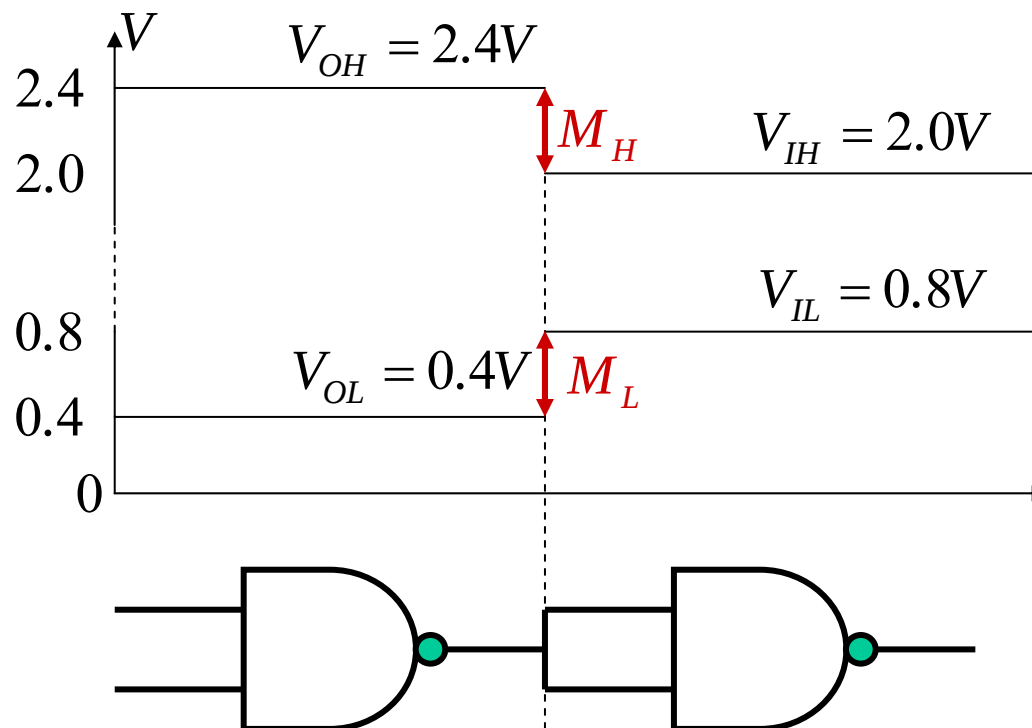
TYPICAL APPLICATION DATA



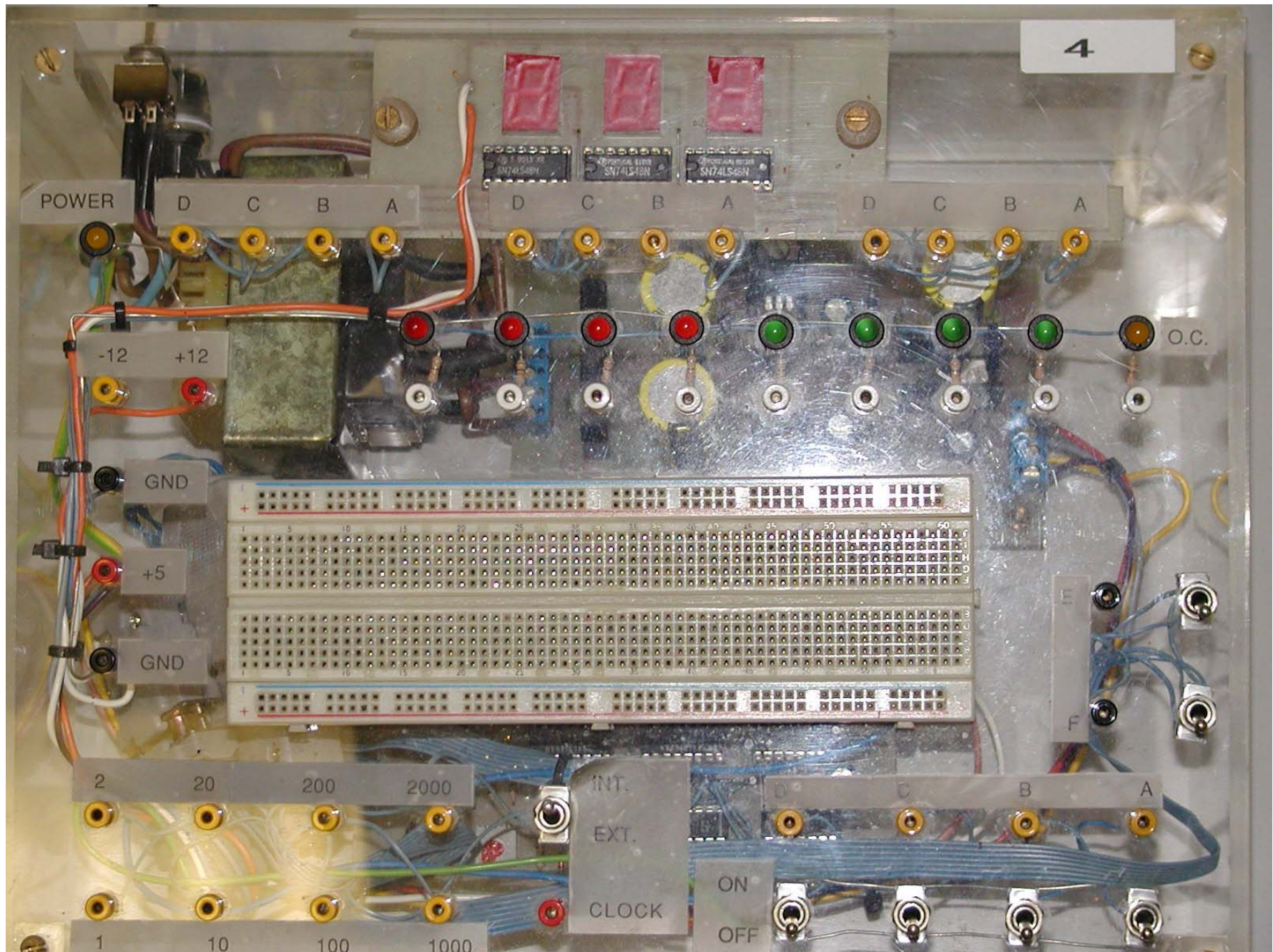
Noi useremo, in generale, integrati (C.I.) della famiglia **TTL (STANDARD?)**

Livelli di ingresso/uscita:

- H: +5 Volts
- L: 0 Volts



1. V_{OH} =minimo valore dello stato alto garantito in uscita
2. V_{OL} =massimo valore dello stato basso garantito in uscita
3. V_{IH} =minimo valore dello stato alto richiesto in ingresso
4. V_{IL} =massimo valore dello stato basso richiesto in ingresso
5. M_H =margine di rumore nello stato basso
6. M_L =margine di rumore nello stato alto



A.A. 2006-07
3° trimestre

Prof. M.Savrié savrie@fe.infn.it
<http://www.fe.infn.it/~savrie/>

45

Uso del "laboratorio logico"

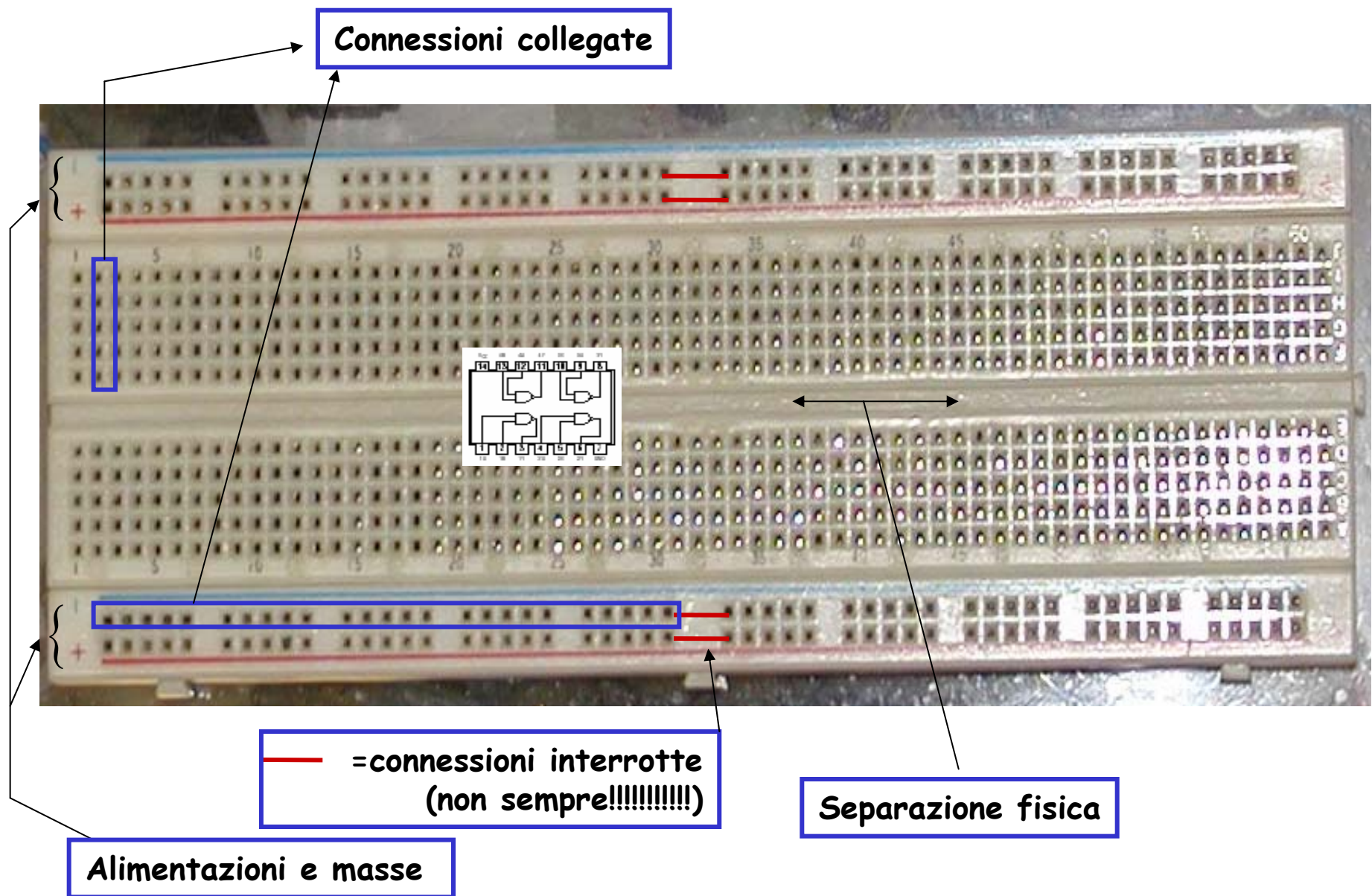
1. V_A = tensione riferita a massa
2. $V_B - V_A$ = tensione "fuori massa"

Le basette ("laboratorio logico") sono dotate di:

- Breadboard
- Alimentazioni: +5V, +12V, -12V
- Clock: 1KHz, (2KHz), 10KHz, (20KHz), 100KHz, (200KHz), 1000KHz, (2000KHz)
- Commutatori (switch)
- Commutatori anti-rimbalo
- Led (active high)
- Led o.c. (active low)
- Display a 7 segmenti

Importante:

Come si usa la breadboard?



Prime esperienze con le porte universali

1. Verifica delle tavole della verità:
 - Porte logiche fondamentali
 - NAND a due ingressi
 - NOR a due ingressi
2. Impiego delle porte NAND E NOR come:
 - Inverter
 - AND, OR con solo porte NAND, NOR
 - Enable, Inhibit
 - Mux, Demux
 - EX-OR con quattro NAND
 - EX-NOR con quattro NOR
 - Comparatore digitale a un bit
 - True/complement
3. Uso di integrati più complessi
 - Complementazione di un numero binario a 4 bit con un 7486 (EXOR come True/Complement)
 - Tavola della verità di una decodifica BCD-7segmenti

Esperienza D-1

- a) Per effettuare operazioni logiche elementari
- b) Per la verifica del **Teorema di De Morgan**
- c) Per controllare il flusso di segnali digitali

Preliminari alle singole prove:

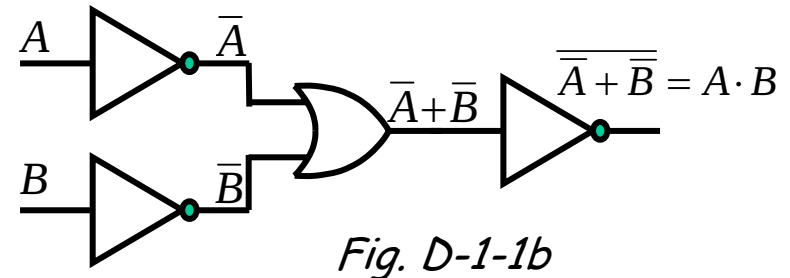
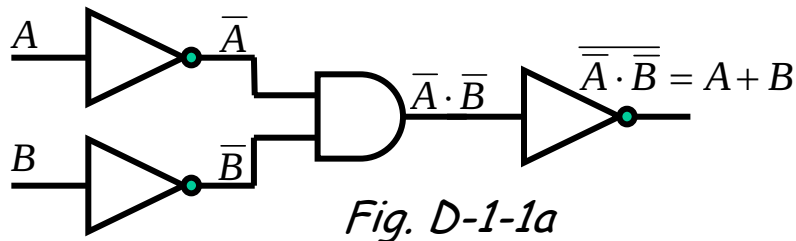
- a) Comprende le prove
 - 1. Verifica delle tavole della verità di NAND (NOR) a due ingressi
 - 2. Impiego di NAND(NOR) come inverter
- b) Comprende le prove
 - 3. Uso di porte NAND (NOR) per realizzare AND(OR)
 - 4. Uso di porte NAND (NOR) per realizzare una porta OR(AND)
- c) Uso di gates per operazioni di:
 - 1. Enable, Inhibit

- a)è banale
- b) Avevamo già visto che dal Teorema di De Morgan:

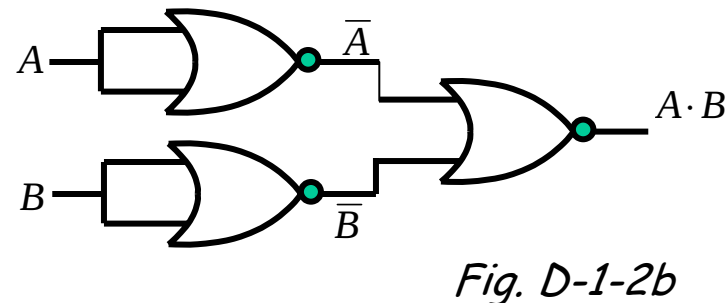
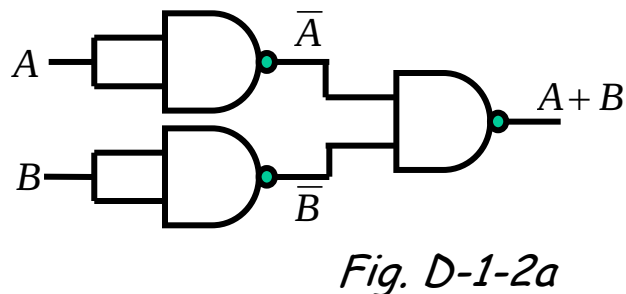
$$\overline{\overline{A} \cdot \overline{B}} = A + B$$

$$\overline{\overline{A} + \overline{B}} = A \cdot B$$

E' sufficiente realizzare il circuito corrispondente al primo membro e verificare che la sua tavola della verità sia uguale a quella del secondo membro.

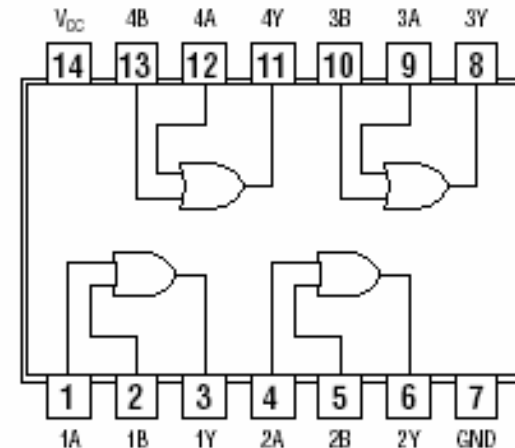


Modificare i circuiti in modo che contengano solo porte NAND o solo porte NOR



Materiale occorrente:

1. Laboratorio logico
2. IC: 7400, 7402, 7404, 7408, 7432
3. Manuale IC

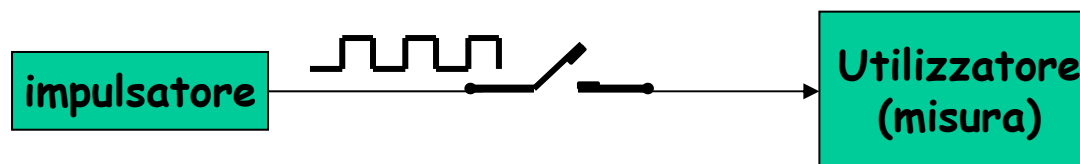


Come si procede:

1. Verificare le tabelle della verità di **TUTTE** le porte
2. Montare gli schemi di cui alle figure:
 - i. D-1-1a e D-1-1b
 - ii. D-1-2a e D-1-2b

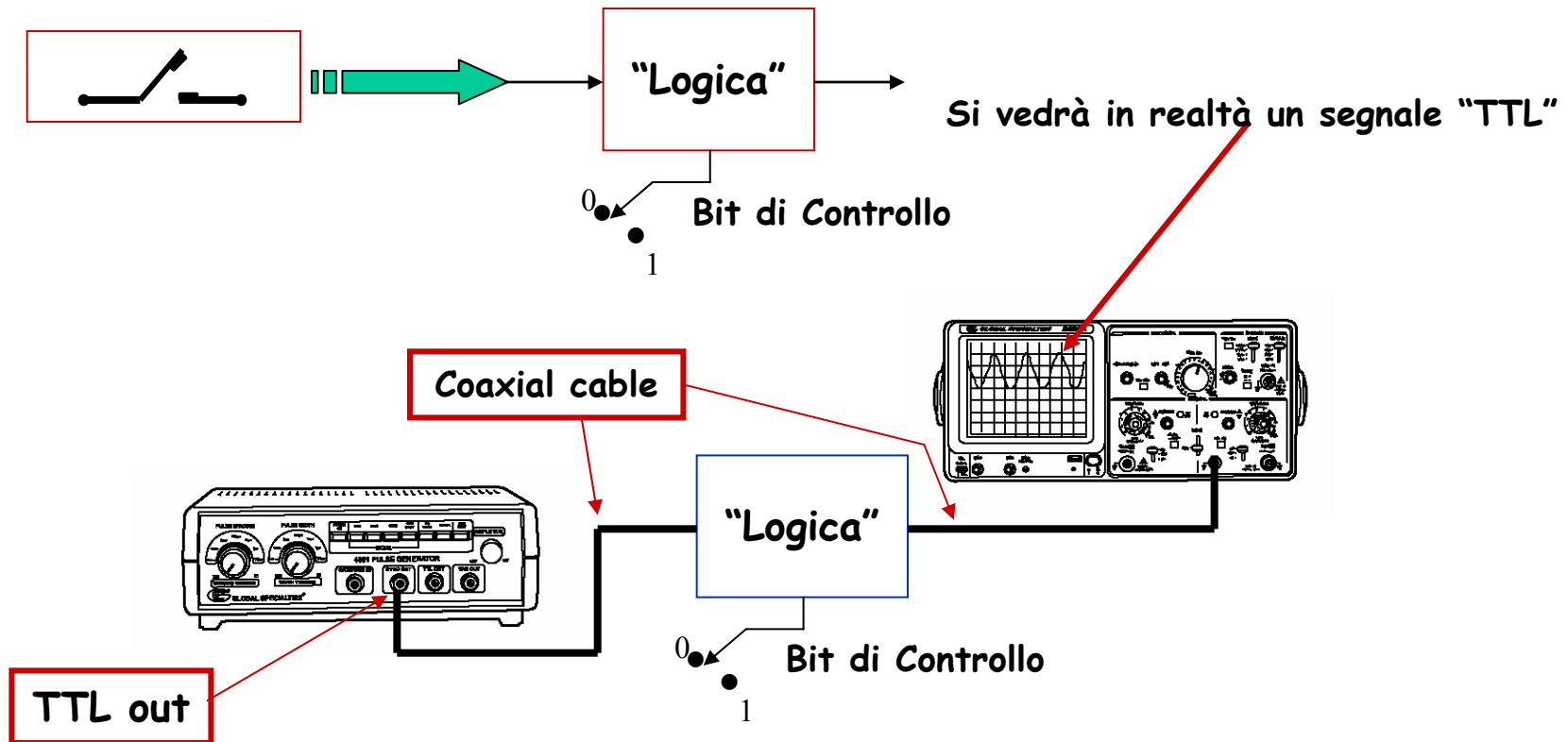
Usare il laboratorio logico che fornisce le alimentazioni per gli IC
i segnali di ingresso (switch) ed i rivelatori di stato di uscita (led)

c) Operazioni di: Enable, Inhibit, True complement, Mux, Demux

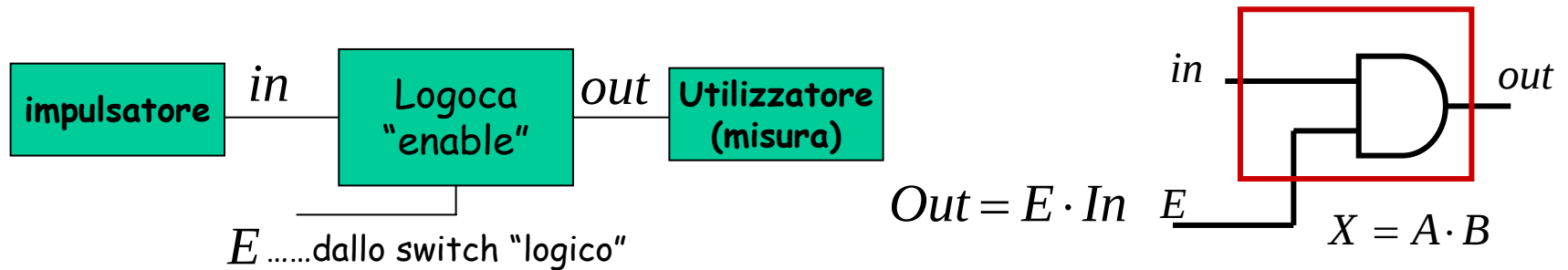


Usare preliminarmente gli switch manuali ed i led dopo avere costruito il blocco "logica" sulla breadboard, poi l' FG ed il CRO (vedi il seguito)

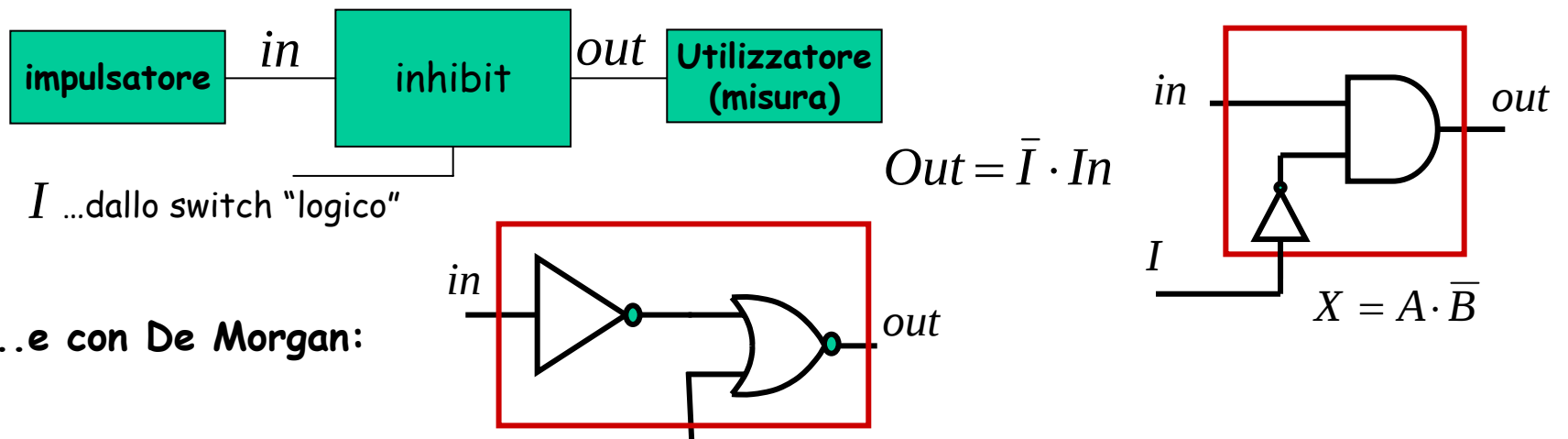
Cerchiamo di sostituire all' interruttore meccanico un meccanismo più sofisticato e pratico:



Enable gate (strobe):



Inhibit gate:



Qualcosa di più su....Livelli e logiche

Livelli:

1. LOGICI: V,F (T,F)
2. FISICI: H,L; (1,0)

Logica:

1. POSITIVA: associazione tra il livello logico **T** e quello fisico **H** e quindi tra **F** e **L**
2. NEGATIVA: associazione tra il livello logico **T** e quello fisico **L** e quindi tra **F** e **H**

Esempio: il '7400 è una quadrupla NAND in logica positiva. Cosa sarebbe in logica negativa?

B	A	$A \cdot B$
F	F	T
F	T	T
T	F	T
T	T	F

tavola della
Verità "logica"

B	A	$A \cdot B$
0	0	1
0	1	1
1	0	1
1	1	0

Logica positiva

Cambiamo logica

B	A	X
1	1	0
1	0	0
0	1	0
0	0	1

Una NOR

Livelli attivi alti:

indica che un certo ingresso produce in uscita l'operazione specificata dal suo nome (Enable, Clear, Reset, Preset.....

o segnala il verificarsi di una situazione indicata (es. $A > B$) quando quel terminale è alto.

0.1

simboli usati nelle tabelle della verità facilmente associabili ai bit di un numero binario e sono da interpretarsi come "equivalenti" a **T** e **F**

Blocchi logici:

1. Combinatori:

lo stato delle uscite in un certo istante, dipende solo dalla combinazione degli ingressi nello stesso istante (a meno del tempo di propagazione del segnale)

2. Sequenziali:

lo stato dell'(e) uscita (e) in un certo istante, dipende(ono), oltre che dalla combinazione degli ingressi nello stesso istante, anche dai valori assunti in precedenza → **memorie**

N.B.

Le funzioni booleane sono **combinatorie**

n bit: 2^n combinazioni possibili

funzioni booleane di n bit: colonne di $m=2^n$ bit
si possono fare 2^m colonne distinte

FUNZIONI LOGICHE DI UNA VARIABILE:

A	F1	F2	F3	F4
0	0	0	1	1
1	0	1	0	1

Es. TTL Texas '7487 True/complement/zero/one

FUNZIONI LOGICHE DI DUE VARIABILI:

B	A	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

**A
N
D**

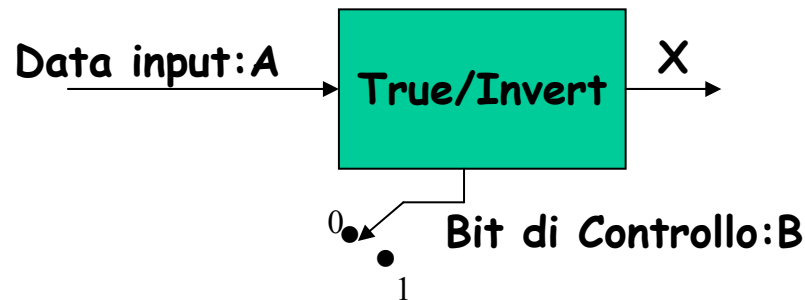
**O
R
N
O
R**

**N
A
N
D**

Esperienza D-2

Comprende le prove:

- Realizzazione di un True/Complement (True/Invert)
- Realizzazione di un XOR(XNOR) con sole NAND e NOR
- Funzione di eguaglianza
- Comparatore digitale a 1 bit
- Mux
- Demux

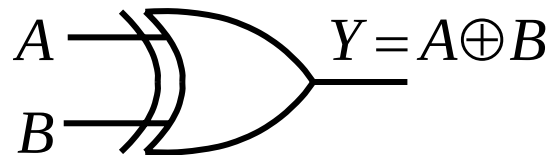


$$X = \begin{cases} A & \text{se } B = 0 \\ \bar{A} & \text{se } B = 1 \end{cases}$$

B	A	X
0	0	0
0	1	1
1	0	1
1	1	0

È un or esclusivo!

EXOR

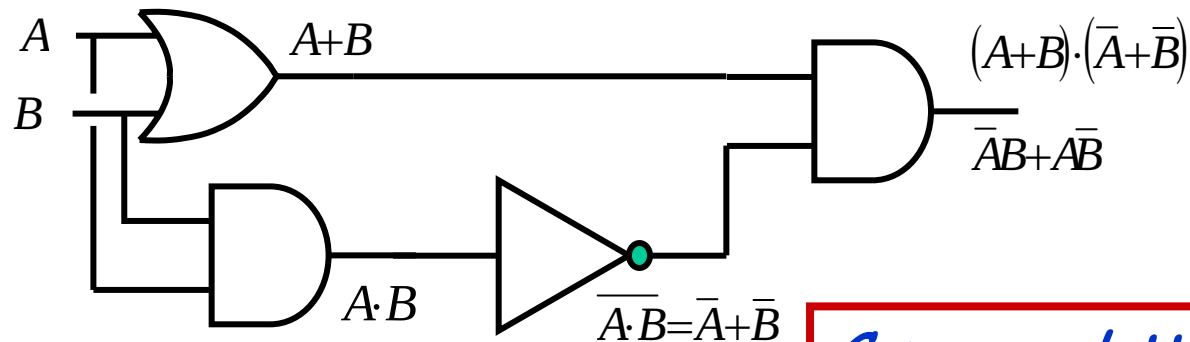


B	A	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

E' uguale alla OR tranne che vale zero se $A=B=1$. Quindi:

è vera se : è vera $(A + B)$
ed è falsa $A \cdot B$

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

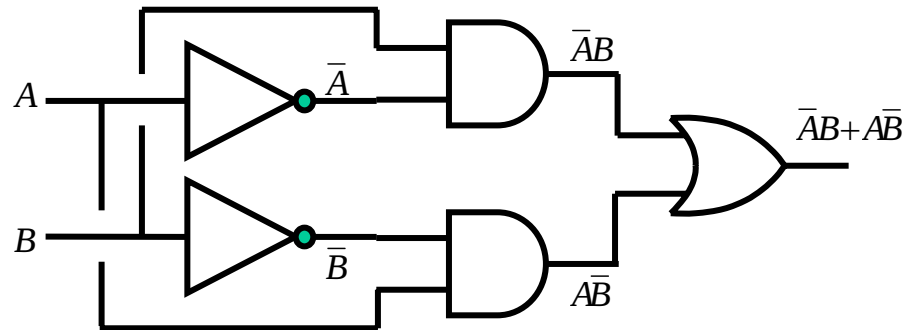


Come prodotti di somme!

Dalla tavola della verità:

B	A	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Come somme di prodotti



Ma sono 5 porte! E si può migliorare.....con: De Morgan

$$\bar{A}B + A\bar{B} = \overline{\overline{\bar{A}B} \cdot \overline{A\bar{B}}}$$

Infatti: $\overline{\bar{A} \cdot B} \cdot \overline{A \cdot \bar{B}} = (\bar{A} + \bar{B}) \cdot (\bar{A} + B) = \bar{A} \cdot \bar{A} + \bar{A} \cdot B + \bar{B} \cdot \bar{A} + \bar{B} \cdot B = \bar{A} \cdot B + \bar{B} \cdot \bar{A}$

Ma : $\overline{\bar{A}B} = \overline{B \cdot (\bar{A} + \bar{B})} = \bar{B} + \overline{(\bar{A} + \bar{B})} = \bar{B} + AB = \overline{B \cdot \bar{A}B}$

Da cui: $\bar{A}B + A\bar{B} = \overline{\overline{\bar{A}B} \cdot \overline{A\bar{B}}}$

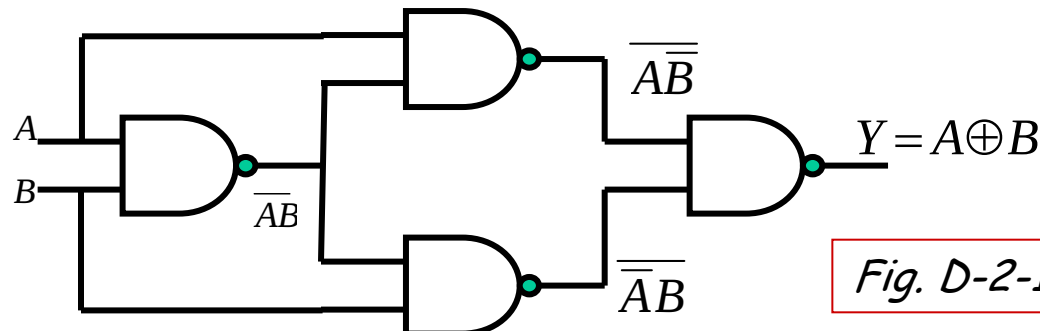
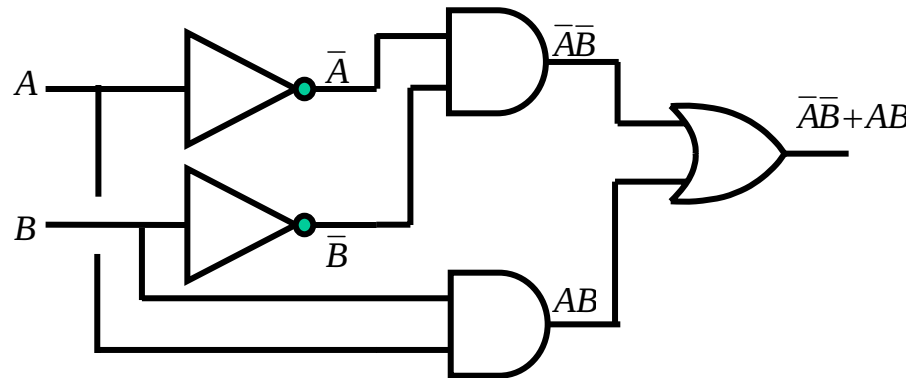


Fig. D-2-1

Oppure per la XNOR:

B	A	$A \oplus B$
0	0	1
0	1	0
1	0	0
1	1	1



Ma sono 5 porte! E si può migliorare.....con: **De Morgan**

La funzione vale 1 solo se $A=B$ (funzione di eguaglianza e complemento della XOR). Per De Morgan il complemento si ottiene: scambiando le AND con le OR e complementando le variabili di ingresso. In questo caso però l'ultima operazione di complementazione non modifica la tavola della verità:

$$XNOR(A,B) = XNOR(\bar{A}, \bar{B})$$

per cui è sufficiente scambiare le AND con le OR:

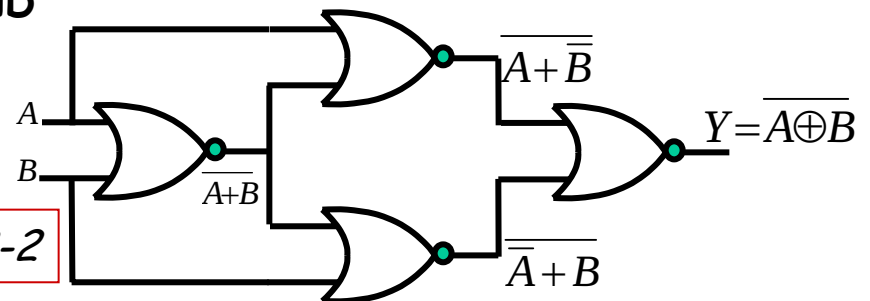


Fig. D-2-2

Realizzazione Pratica:

1. usare un I.C. 7400 (quadrupla NAND a due ingressi) per realizzare lo schema (fig. D-2-1) e ricavarne la tavola della verità;

B	A	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Se si usa una delle variabili (es. B) come bit di controllo:

$$\begin{aligned} B = 0 &\Rightarrow Out = A \\ B = 1 &\Rightarrow Out = \bar{A} \end{aligned}$$

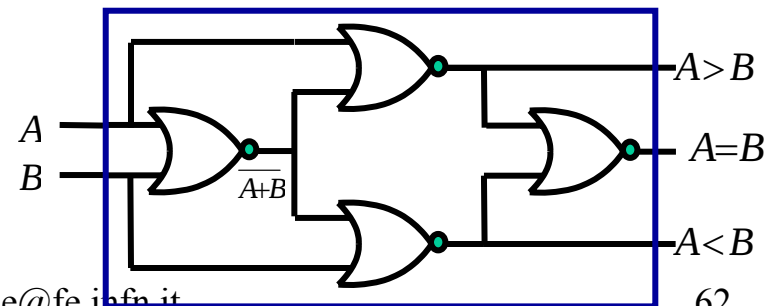
True/complement

2. usare un I.C. 7402 (quadrupla NOR a due ingressi) per realizzare lo schema (fig. D-2-2) e ricavarne la tavola della verità;

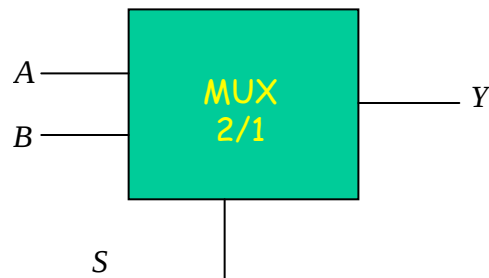
B	A	$A \oplus B$
0	0	1
0	1	0
1	0	0
1	1	1

- E' vera quando sono diversi gli ingressi
- la porta di uscita (NOR nella fig.D-2-2) è falsa quando o l'uno o l' altro dei due ingressi sono veri -> i due ingressi devono rappresentare $A > B$ e $A < B$:

Blocco comparatore
A 1 bit



Multiplexers e



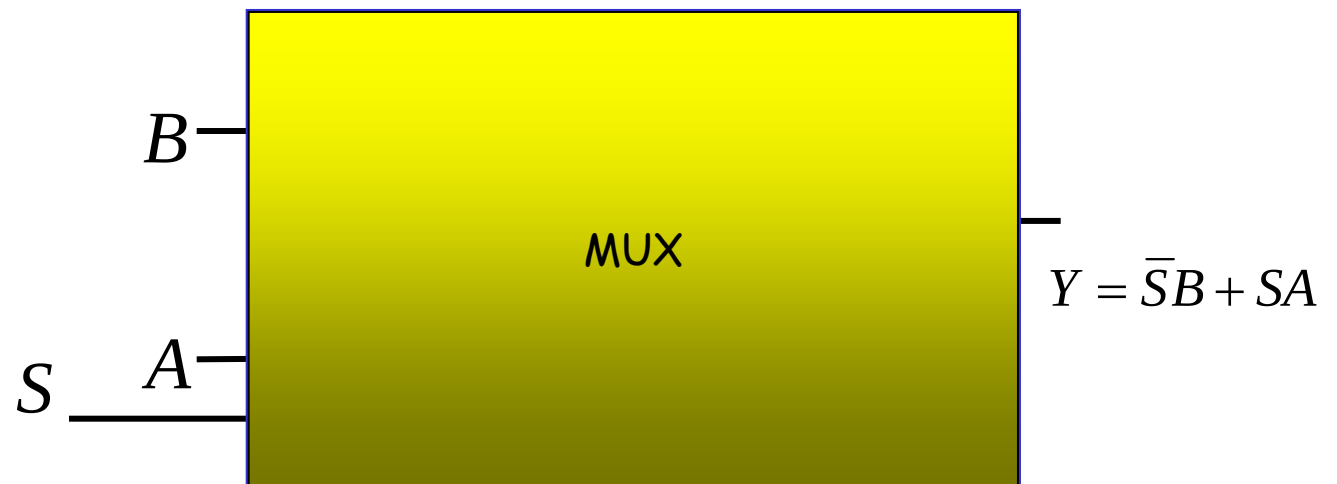
S	A	B	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

S	A	B	Y
0	x	x	B
1	x	x	A

COME PREVEDIBILE ABBIAMO UNA SOLA FUNZIONE LOGICA:

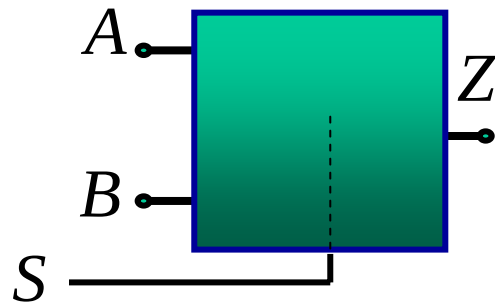
$$Y = \overline{A}\overline{B}\overline{S} + A\overline{B}\overline{S} + \overline{A}B\overline{S} + AB\overline{S}$$

$$Y = \overline{S}B + SA$$



Multiplexer (italiano: selettore):

esempio: 2 su 1



S	B	A	Z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

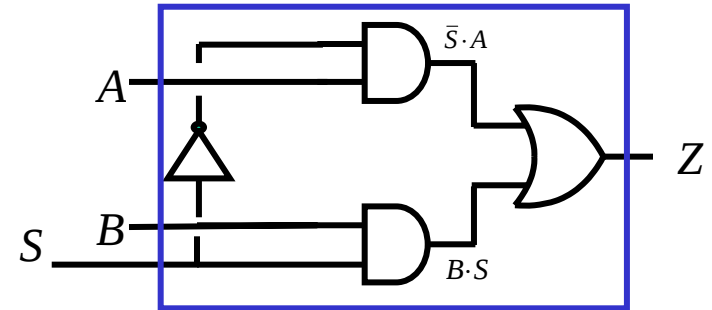
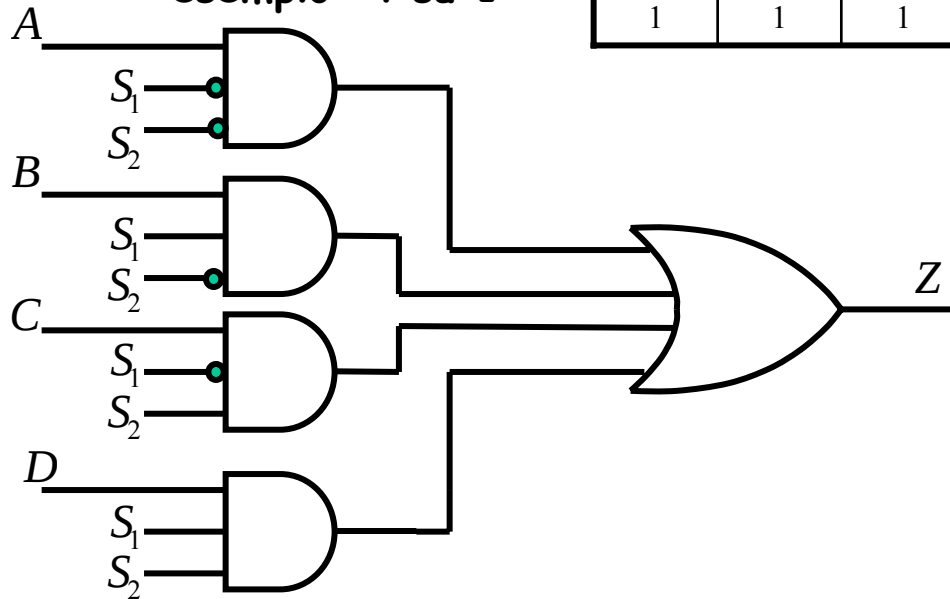


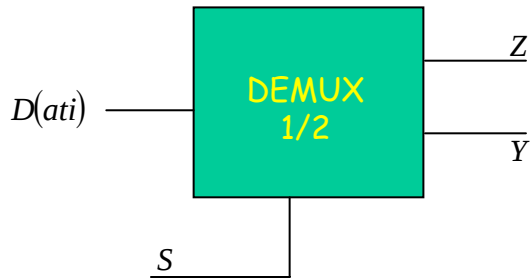
Fig. D-2-3

esempio: 4 su 1



S1	S2	Z
0	0	A
0	1	C
1	0	B
1	1	D

..... Demultiplexers



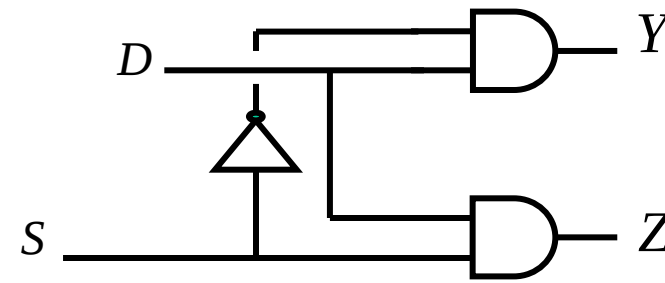
S	D	Z	Y
0	0	0	0
0	1	0	1
1	0	0	0
1	1	1	0

S	D	Z	Y
0	x	0	D
1	x	D	0

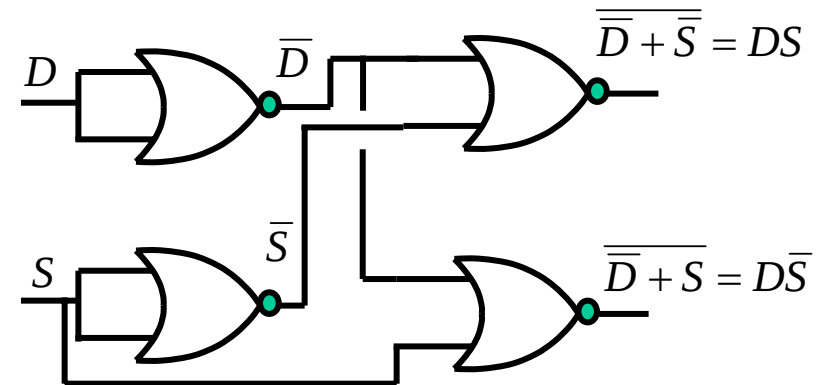
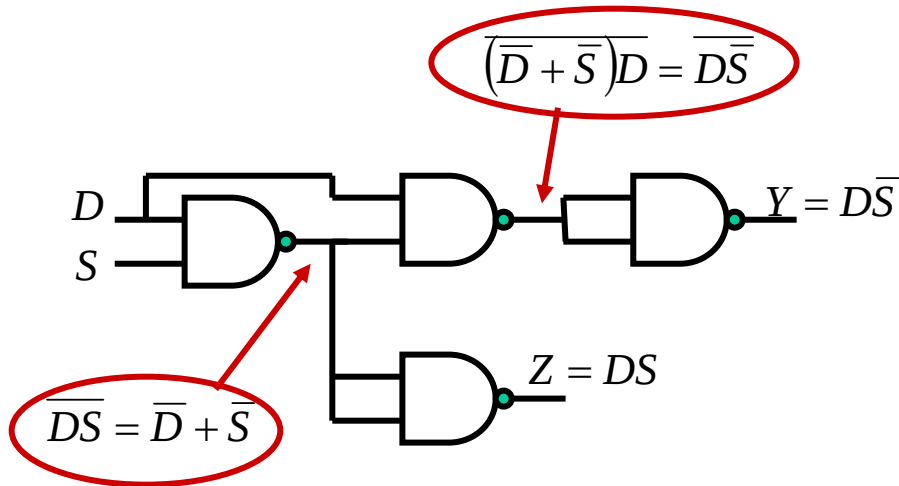
COME PREVEDIBILE ABBIAMO
DUE FUNZIONI LOGICHE:

$$Z = SD$$

$$Y = \bar{S}D$$

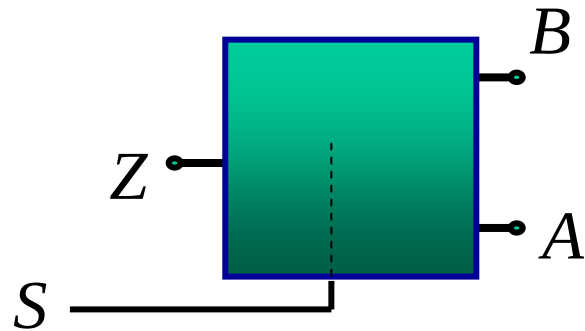


Con le porte universali:



Demultiplexer:

esempio: 1 su 2



S	D	y	Z
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

S	D	y	Z
0	X	D	0
1	X	0	D

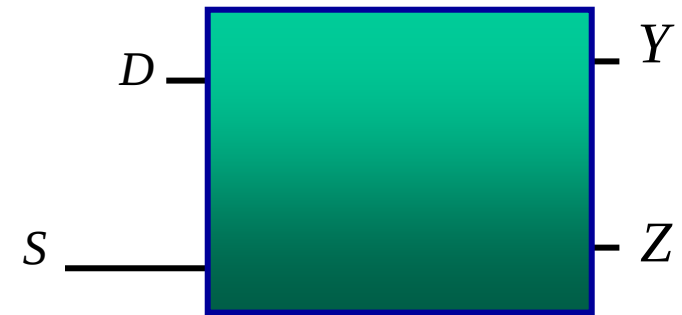
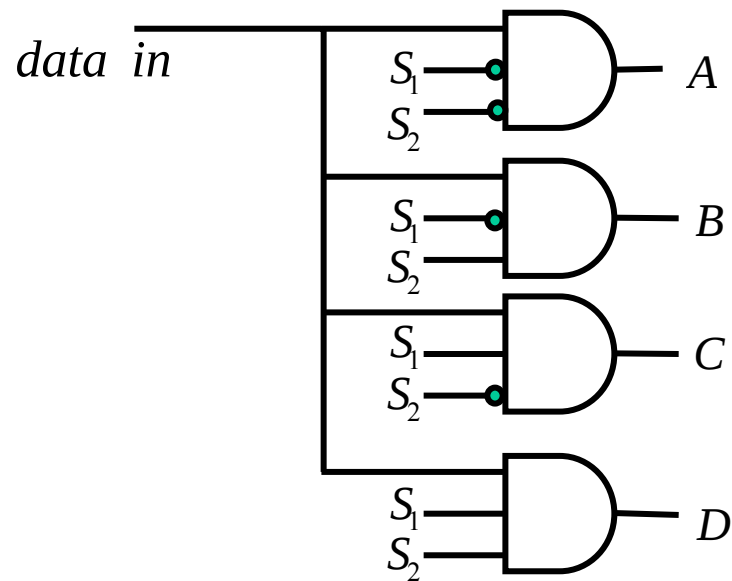


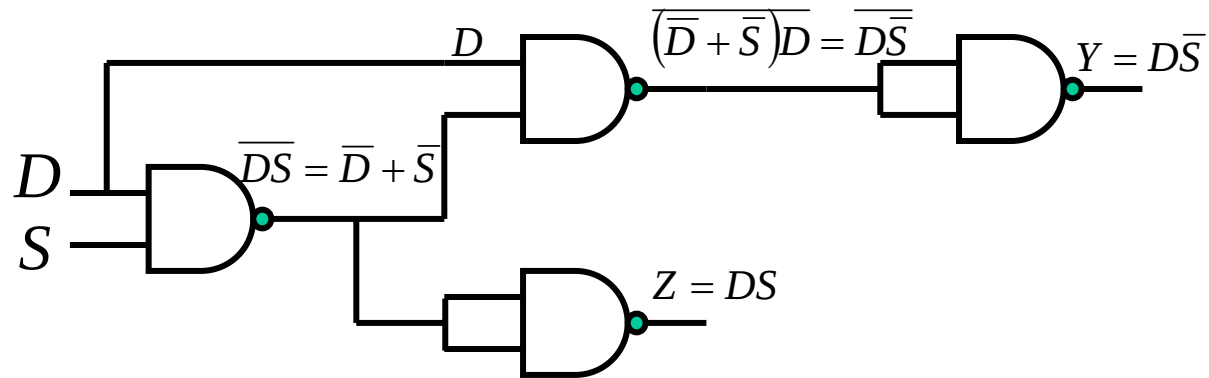
Fig. D-2-4

esempio: 1 su 4

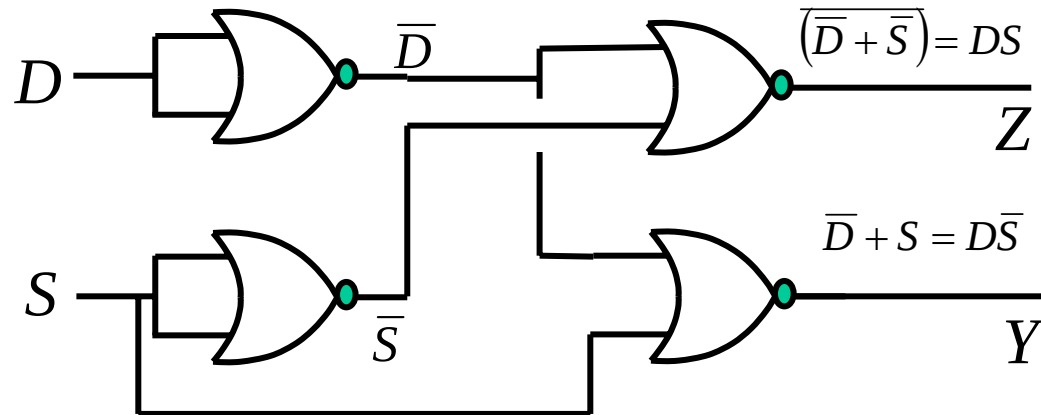


S1	S2	OUT
0	0	A
0	1	B
1	0	C
1	1	D

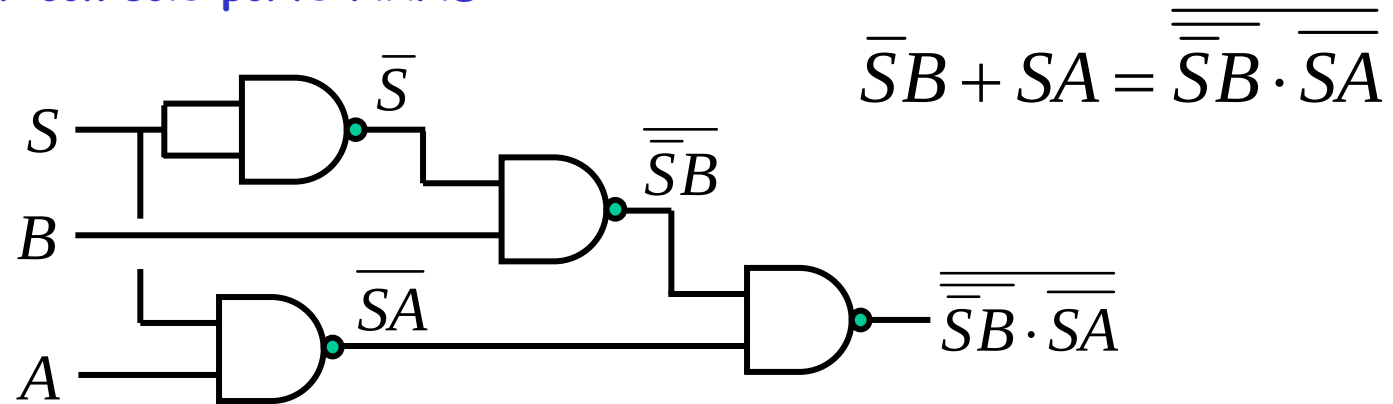
Demultiplexer con sole porte NAND:



Demultiplexer con sole porte NOR:

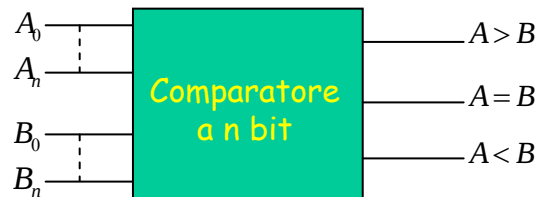


Multiplexer con sole porte NAND:



Multiplexer con sole porte NOR: più di 4 → non è utile adesso

Comparatore



A	B	A<B	A=B	A>B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

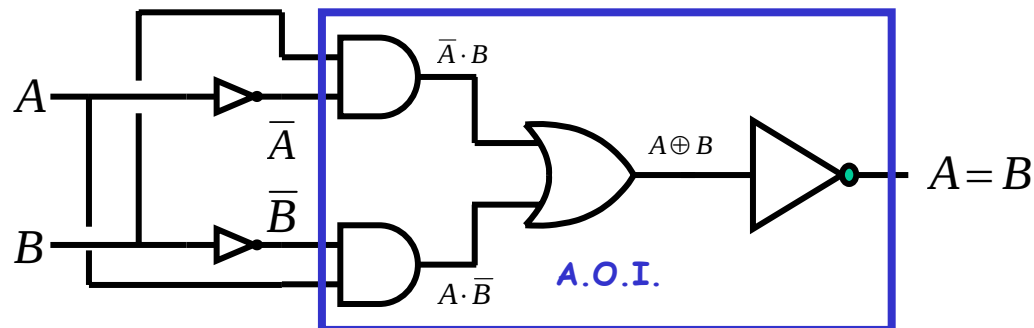
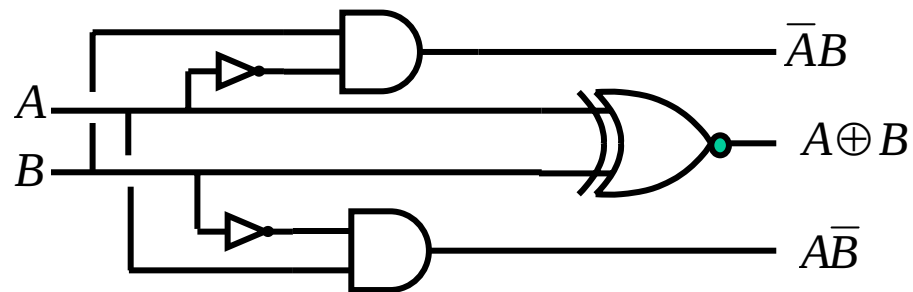
Partiamo dal caso più semplice a 1 bit:

COME PREVEDIBILE ABBIAMO TRE FUNZIONI LOGICHE:

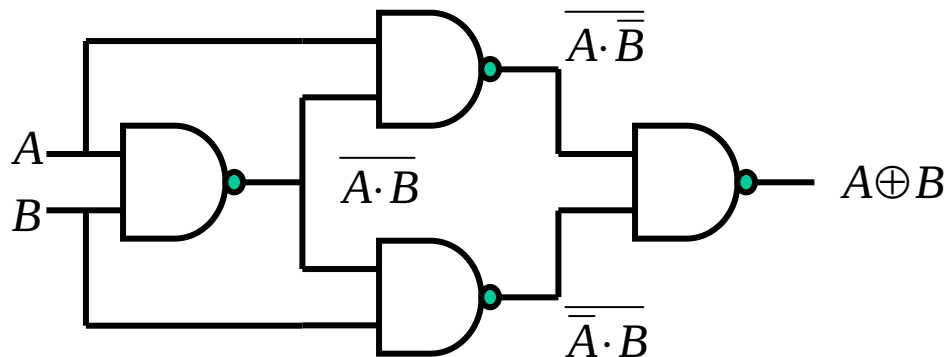
$$A < B = \bar{A} \cdot B$$

$$A > B = A \cdot \bar{B}$$

$$A = B = \bar{A} \cdot \bar{B} + AB$$

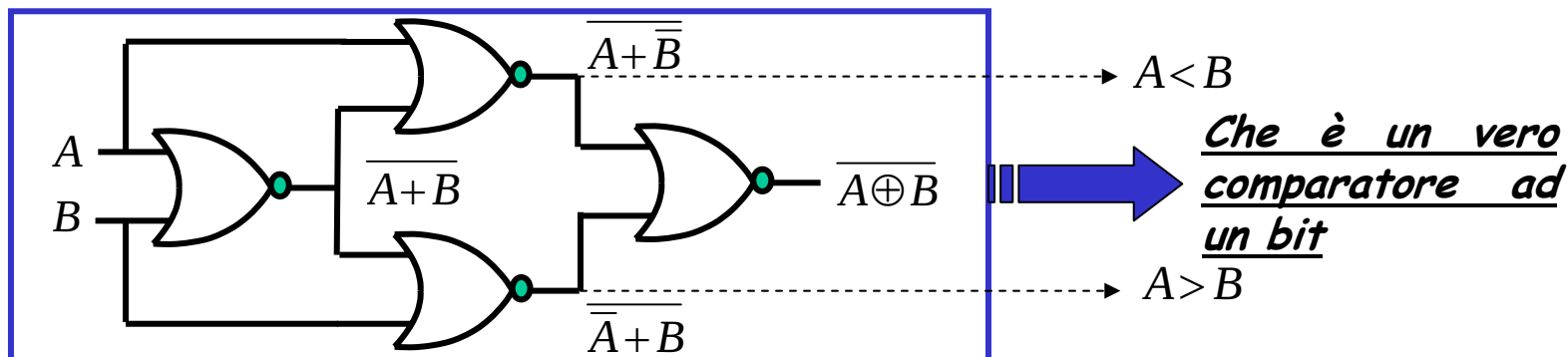


Avevamo già visto come fare un **EXOR** con 4 porte **NAND**



Ma è anche più immediato se si realizza mediante **NOR** infatti per **De Morgan**:

$$\overline{A+B} = \bar{A} \cdot \bar{B} \quad \overline{\bar{A}+B} = A \cdot \bar{B}$$



.....lo riprenderemo più avanti!

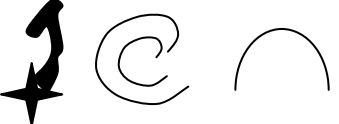
Condotta Pratica:

1. **Assemblare il circuito MUX 2 su 1 (fig D-2-3)**
 - **Realizzare il MUX con sole NAND**
2. **Asseblare il circuito Demux 1 su 2**
 - **Realizzarlo anche con sole NAND o NOR**
3. **Provare ciascuno dei circuiti con gli interruttori manuali**
4. **Provare ciascun circuito con generatore di Funzioni (F.G.) e oscilloscopio (C.R.O.)**
5. **provare un comparatore ad 1 bit**
6. **Provare anche 74153 (dual quad-in MUX)**
7. **Provare anche 74154 (4in-16 out)**
8. **Provare un 7485 (comparatore a 4 bit)**

Sistemi di Numerazione

- Numero
 - concetto non primitivo
 - serve per “quantificare” la realtà
- Sistema di numerazione:
 - insieme finito di simboli
 - simboli organizzati in sequenze secondo “regole”

Il sistema più antico è (forse) quello egizio ed ha circa 5000 anni. E' di tipo decimale con simboli ripetuti per i multipli di una stessa quantità.

1		100	
2		231	
10		1000	
11		1101	
20			

- **I Sumeri avevano l' unità numerica fondamentale che corrisponde al nostro 60**

nostra misurazione degli angoli?

Non hanno lo zero ed i simboli sono posizionali

- lo zero introdotto forse nella civiltà indiana e poi arriva in Europa portato dagli Arabi
- I simboli da noi usati oggi (indo-Arabi) risalgono al X secolo
- I numeri frazionari arrivano solo nel XVI secolo
- Il punto decimale viene introdotto verso la metà del XVII secolo

Basi: numero di simboli usati nella numerazione

Posizionale: sistema in cui il valore di associato ad ogni simbolo dipende dalla sua posizione nella “stringa”

Peso: il fattore per cui il simbolo (numero) deve essere moltiplicato per potere essere confrontato con gli altri simboli (numeri): potenza ad esponente variabile della base del sistema di numerazione

Sistema decimale:

- è in base 10
- 10 simboli: 0-9
- è posizionale

Esempio: 4518,23

4	5	1	8,	2	3
10^3	10^2	10^1	10^0 ,	10^{-1}	10^{-2}

Che significa:

4000+500+10+8+0,2+0,03

Sistema binario:

- è in base 2
- 2 simboli: 0,1
- è posizionale

Esempio: 1001,01

1	0	0	1,	0	1
2^3	2^2	2^1	2^0 ,	2^{-1}	2^{-2}

Che significa:

8+0+0+1+0+0.25

Come si passa da un sistema all' altro?

✓ **Binario -> Decimale**

$$(1100101)_2 =$$

$$1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = (101)_{10}$$

✓ **Decimale -> Binario**

$$(37)_{10}$$

$$37 \div 2$$

$$18 \div 2$$

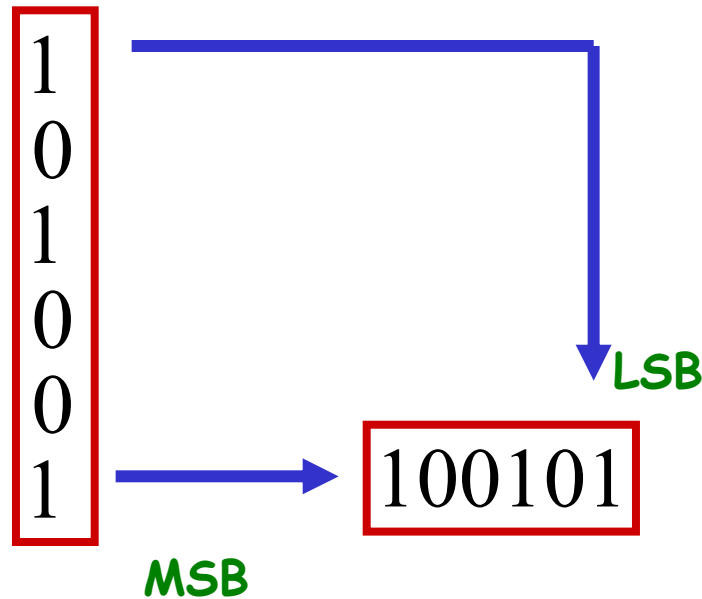
$$9 \div 2$$

$$4 \div 2$$

$$2 \div 2$$

$$1 \div 2$$

$$0$$



Sistema ottale:

- è in base 8
- 8 simboli: 0-7
- è posizionale

Esempio: $(514,23)_8$

5	1	4,	2	3
8^2	8^1	8^0 ,	8^{-1}	8^{-2}

Esempio: $(456)_8$

1	0	0	1	0	1	1	1	0
---	---	---	---	---	---	---	---	---

Sistema esadecimale:

- è in base 16
- 16 simboli: 0-9,A,B,C,D,E,F
- è posizionale

Esempio: $(3AFF9)_{16}$

3	10	15	15	9
16^4	16^3	16^2	16^1	16^0

Esempio: $(B7F)_{16}$

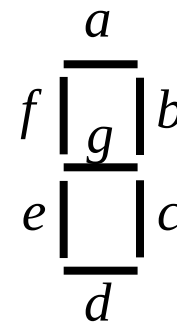
1	0	1	1	0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---

Esempio paratiko di progetto (parziale)

Convertitore BCD-7 segmenti

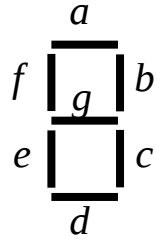
BCD: binary coded decimal

7 segmenti: è un tipo di display



Come si procede:

1. Dobbiamo rappresentare i numeri da 0 a 9:
di quanti bit abbiamo bisogno?
2. Quale (i) è (sono) la (le) funzione (i) logica?
3. Come si costruisce (ono)?



															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

'46A, '47A, 'LS47 FUNCTION TABLE (T1)

DECIMAL OR FUNCTION	INPUTS						$\overline{\text{BI}}/\text{RBO}^\dagger$	OUTPUTS							NOTE
	$\overline{\text{LT}}$	$\overline{\text{RBI}}$	D	C	B	A		a	b	c	d	e	f	g	
0	H	H	L	L	L	L	H	ON	ON	ON	ON	ON	ON	OFF	1
1	H	X	L	L	L	H	H	OFF	ON	ON	OFF	OFF	OFF	OFF	
2	H	X	L	L	H	L	H	ON	ON	OFF	ON	ON	OFF	ON	
3	H	X	L	L	H	H	H	ON	ON	ON	ON	OFF	OFF	ON	
4	H	X	L	H	L	L	H	OFF	ON	ON	OFF	OFF	ON	ON	
5	H	X	L	H	L	H	H	ON	OFF	ON	ON	OFF	ON	ON	
6	H	X	L	H	H	L	H	OFF	OFF	ON	ON	ON	ON	ON	
7	H	X	L	H	H	H	H	ON	ON	ON	OFF	OFF	OFF	OFF	
8	H	X	H	L	L	L	H	ON	ON	ON	ON	ON	ON	ON	
9	H	X	H	L	L	H	H	ON	ON	ON	OFF	OFF	ON	ON	
10	H	X	H	L	H	L	H	OFF	OFF	OFF	ON	ON	OFF	ON	
11	H	X	H	L	H	H	H	OFF	OFF	ON	ON	OFF	OFF	ON	
12	H	X	H	H	L	L	H	OFF	ON	OFF	OFF	OFF	ON	ON	
13	H	X	H	H	L	H	H	ON	OFF	OFF	ON	OFF	ON	ON	
14	H	X	H	H	H	L	H	OFF	OFF	OFF	ON	ON	ON	ON	
15	H	X	H	H	H	H	H	OFF	OFF	OFF	OFF	OFF	OFF	OFF	
BI	X	X	X	X	X	X	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	2
RBI	H	L	L	L	L	L	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3
LT	L	X	X	X	X	X	H	ON	ON	ON	ON	ON	ON	ON	4

H = high level, L = low level, X = irrelevant

- NOTES:
1. The blanking input ($\overline{\text{BI}}$) must be open or held at a high logic level when output functions 0 through 15 are desired. The ripple-blanking input ($\overline{\text{RBI}}$) must be open or high if blanking of a decimal zero is not desired.
 2. When a low logic level is applied directly to the blanking input ($\overline{\text{BI}}$), all segment outputs are off regardless of the level of any other input.
 3. When ripple-blanking input ($\overline{\text{RBI}}$) and inputs A, B, C, and D are at a low level with the lamp test input high, all segment outputs go off and the ripple-blanking output ($\overline{\text{RBO}}$) goes to a low level (response condition).
 4. When the blanking input/ripple blanking output ($\overline{\text{BI}}/\text{RBO}$) is open or held high and a low is applied to the lamp-test input, all segment outputs are on.

$^\dagger \overline{\text{BI}}/\text{RBO}$ is wire AND logic serving as blanking input ($\overline{\text{BI}}$) and/or ripple-blanking output ($\overline{\text{RBO}}$).

Consideriamo il segmento e

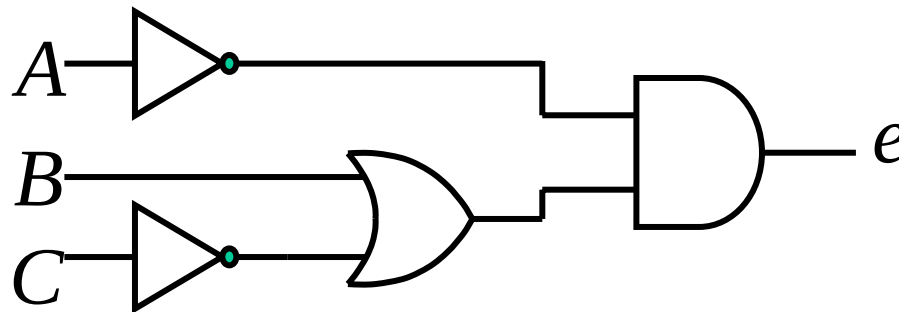
$$e = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD$$

.....e semplifichiamo la funzione!

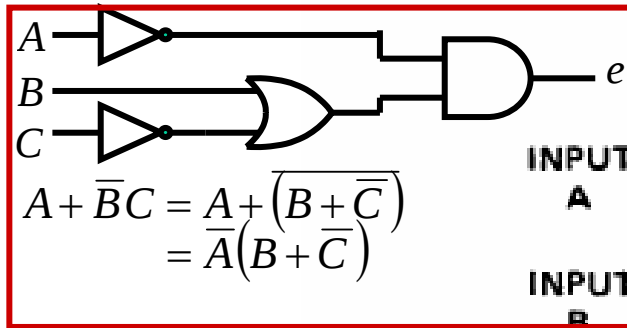
$$\begin{aligned} e &= \overline{A}\overline{C}\overline{D}(B + \overline{B}) + \overline{A}BC(D + \overline{D}) + \overline{A}\overline{C}D(B + \overline{B}) \\ &= \overline{A}\overline{C}\overline{D} + \overline{A}BC + \overline{A}\overline{C}D = \overline{A}\overline{C}(\overline{D} + D) + \overline{A}BC \end{aligned}$$

.....poi si ricorre ad un trucco:

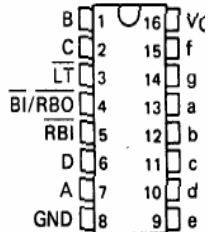
$$\begin{aligned} e &= \overline{A}\overline{C} + \overline{A}BC = \overline{A}\overline{C}(1 + B) + \overline{A}BC \\ &= \overline{A}\overline{C} + \overline{A}\overline{C}B + \overline{A}BC = \overline{A}\overline{C} + \overline{A}B = \boxed{\overline{A}(\overline{C} + B)} \end{aligned}$$



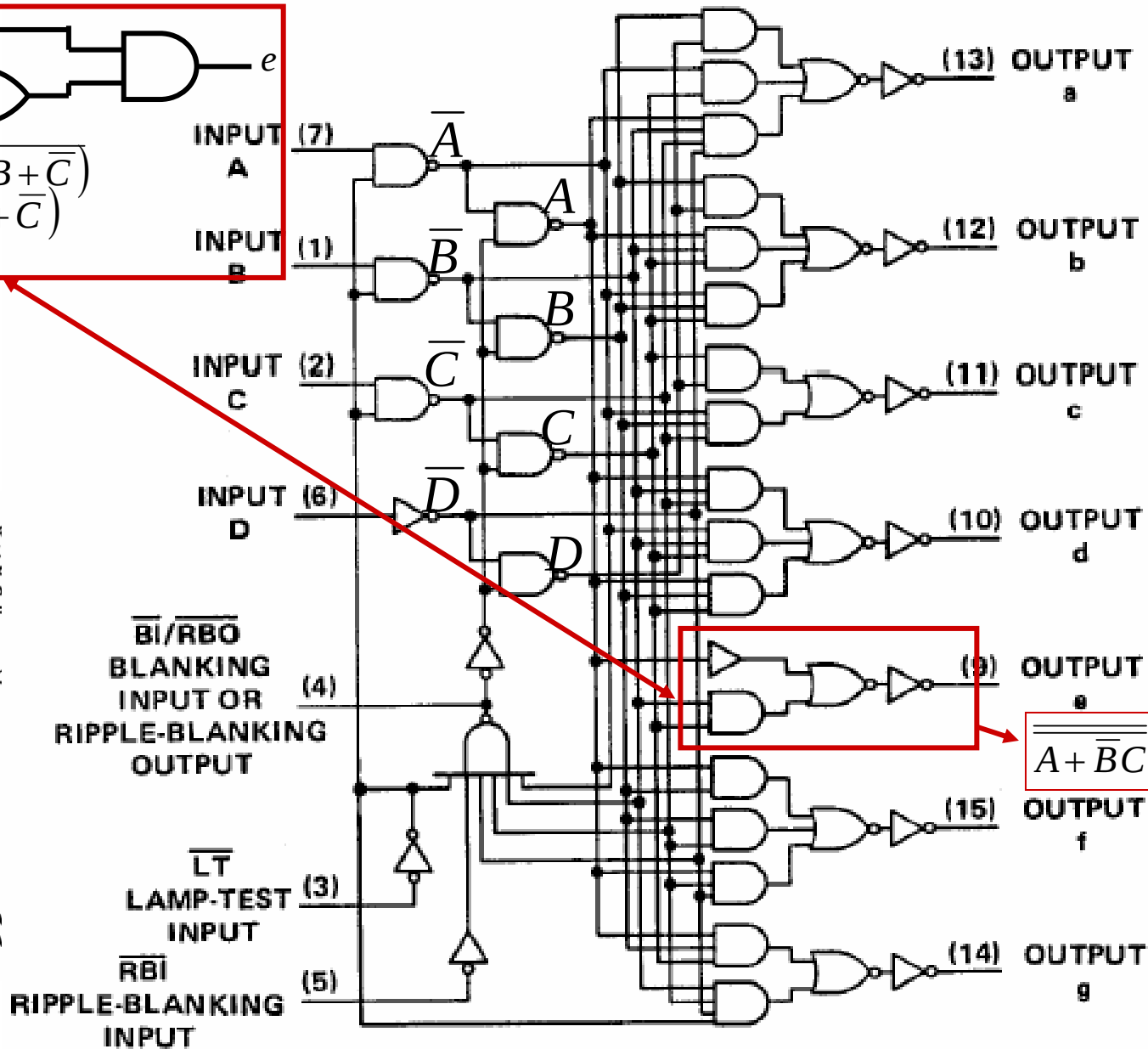
'46A, '47A, 'LS47

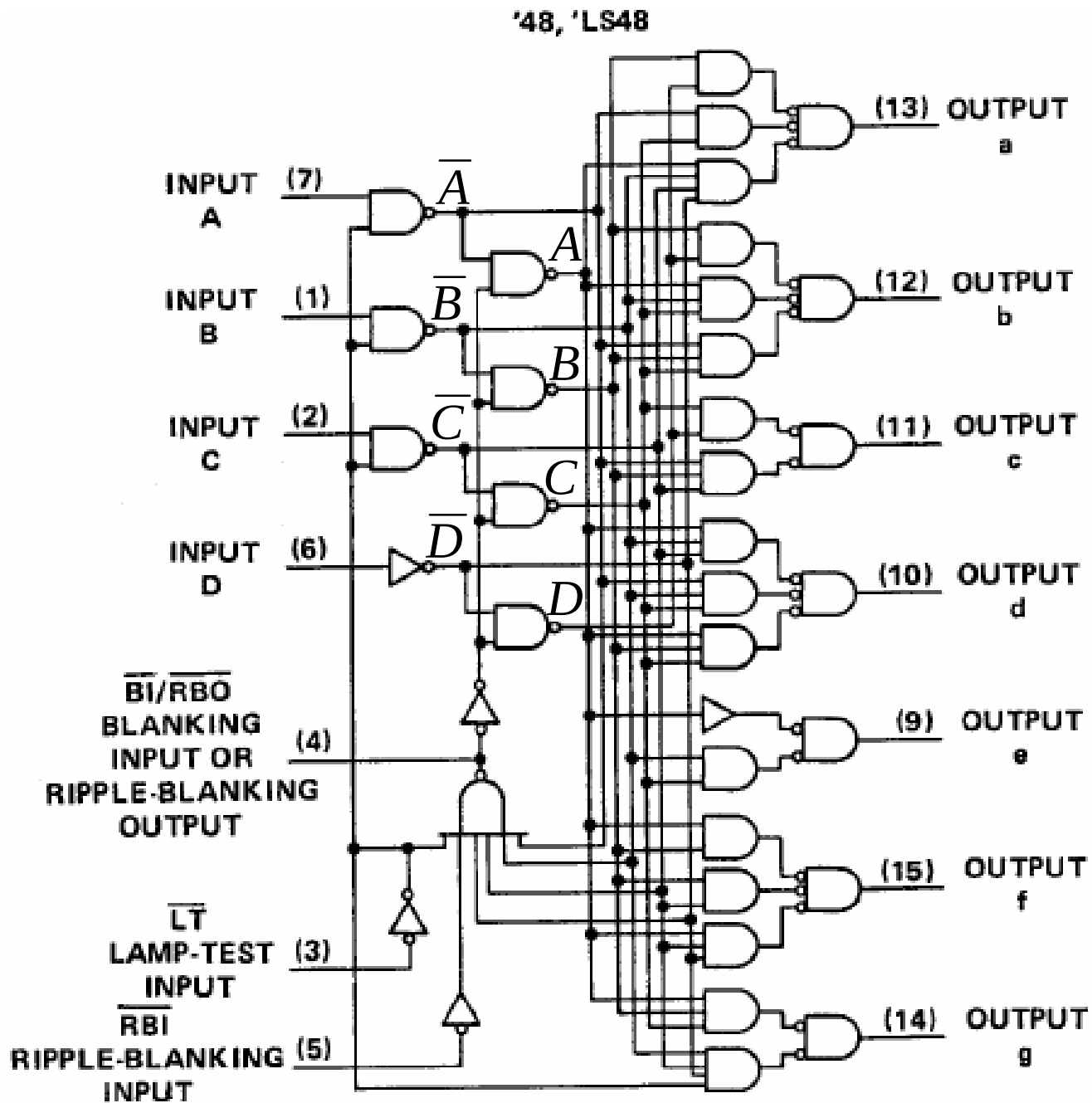


SN5446A, SN5447A, SN54LS4
 SN54LS48 ... J PACK
 SN7446A, SN7447A
 SN7448 ... N PACKA
 SN74LS47, SN74LS48 ... D OF
 (TOP VIEW)



SN54LS49 ... J OR W PA
 SN74LS49 ... D OR N PA
 (TOP VIEW)



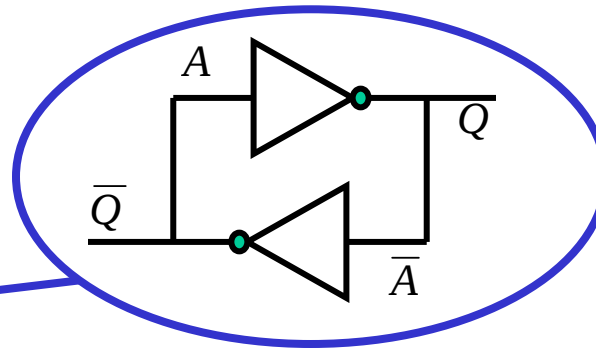


Fino ad ora abbiamo considerato solo: **LOGICA COMBINATORIA**
che non ha "memoria"

Sappiamo che esiste anche una: **LOGICA SEQUENZIALE**
che si "ricorda" della storia precedente.

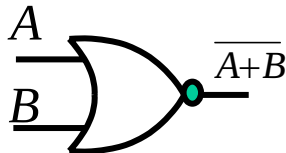
La logica sequenziale si basa sul concetto di bistabilità (multiviratori):

1. Circuiti **astabili**
2. Circuiti **monostabili**
3. Circuiti **bistabili**

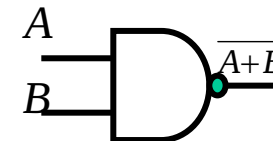


Ma quali problemi ha?

e se facciamo gli invertitori con **NAND** e **NOR**?

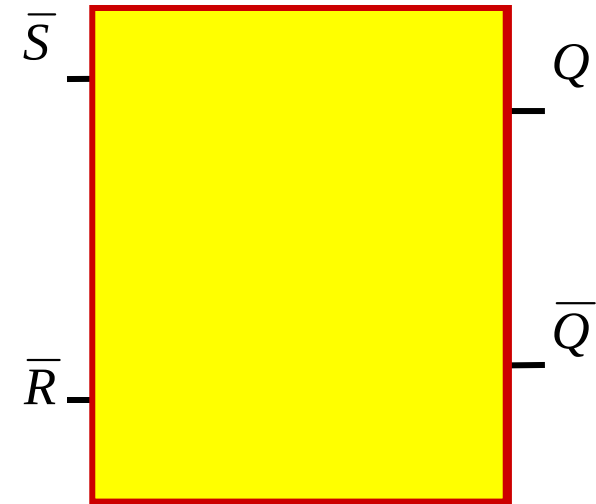
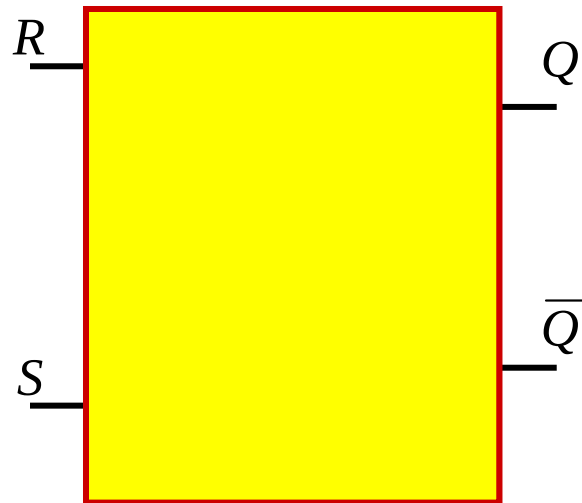


B	A	$\overline{A+B}$
L	L	H
L	H	L
H	L	L
H	H	L



B	A	$\overline{A \cdot B}$
L	L	H
L	H	H
H	L	H
H	H	L

Come si interpretano?



Tipo
set-reset

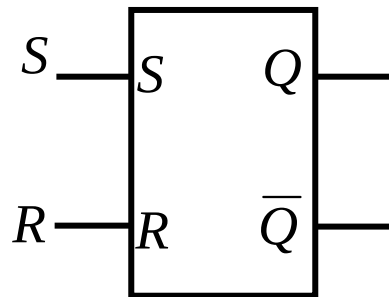
*Questi circuiti sono in grado di
memorizzare 1 bit e si chiamano:*

Tipo
nonset-nonreset

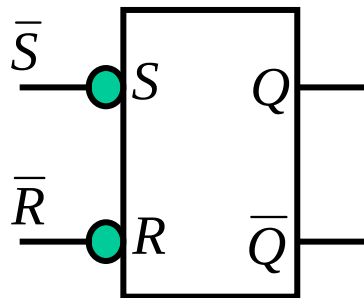
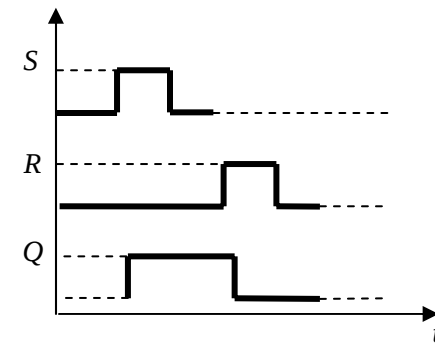
FLIP FLOP

NB

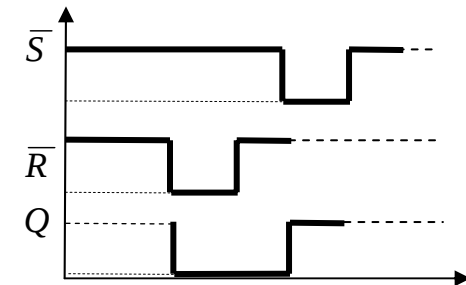
Hanno comportamento bistabile: sotto l' effetto di uno stimolo di comando in ingresso (S,R) generano uno stato stabile in uscita (Q, $\bar{Q}$) che permane anche una volta cessato il comando



S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	<i>n.p.</i>



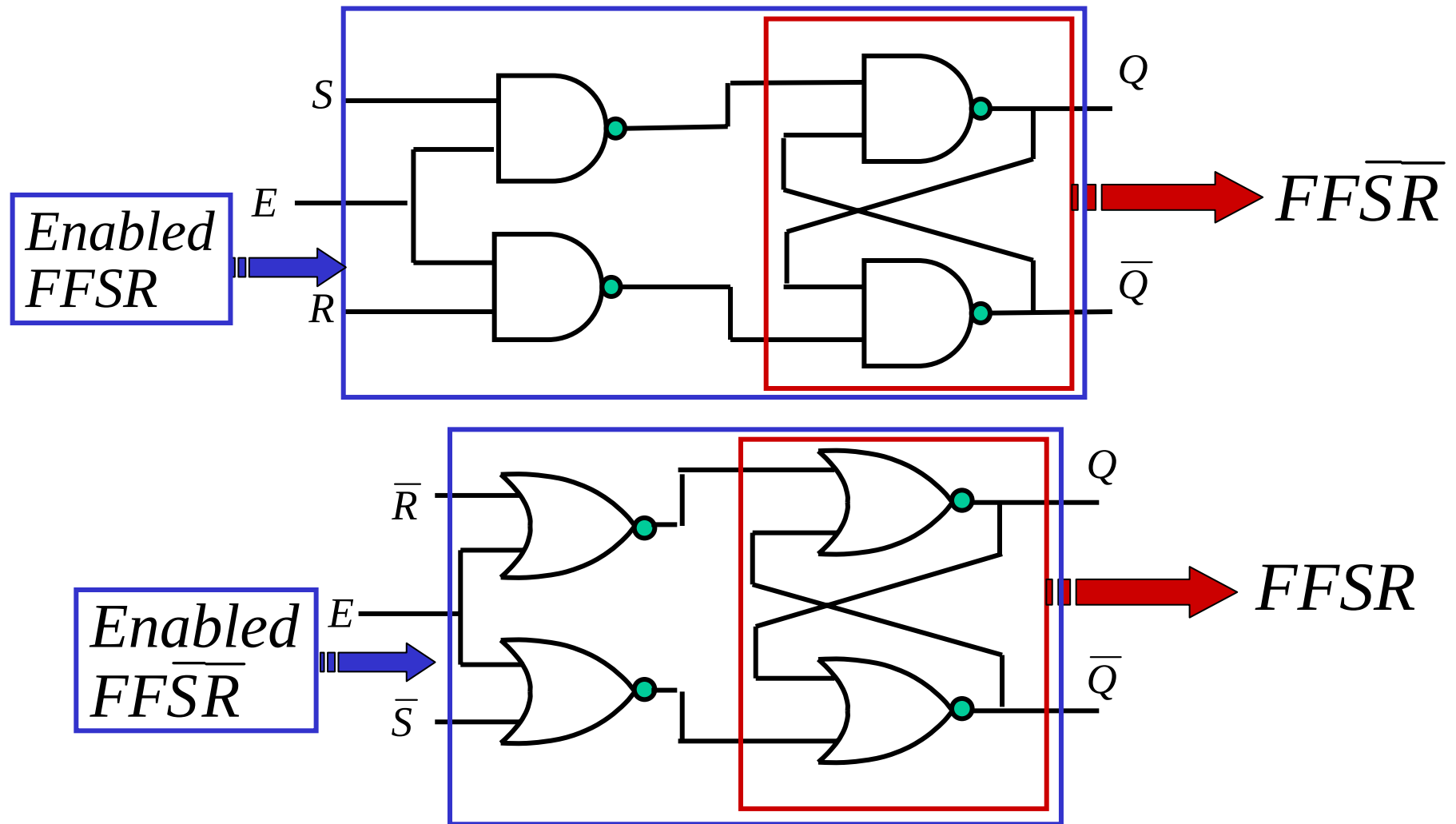
S	R	Q_{n+1}
1	1	Q_n
0	1	1
1	0	0
0	0	<i>n.p.</i>

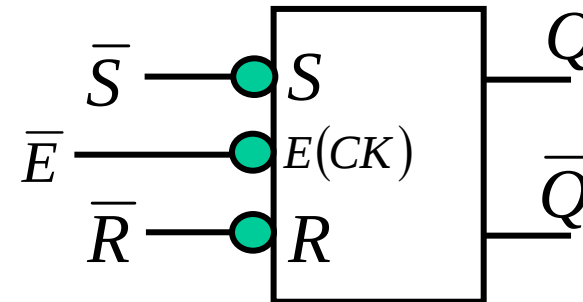
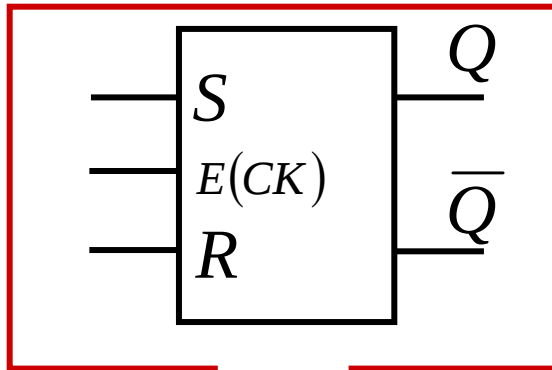


Attenzione:

1. Quali associamo ai circuiti realizzati con sole NOR e sole NAND?
2. si attiva un solo ingresso per volta
3. Il circuito è "sincronizzabile" (abilitazione)
4. In uno stesso FF si possono avere ingressi attivi alti e attivi bassi

FLIP FLOP..... con abilitazione





E	S	R	Q_{n+1}
0	X	X	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	$n.p.$

← *Il FF è bloccato*



← *Con E abilitato funziona come
Un normale SR*

← *indeterminato*

Se usiamo l' ENABLE:

1. Per bloccare o meno il FF: **comportamento banale**
2. Per configurare i dati (S,R) per avere una certa uscita ($Q, \bar{Q}$) in un certo istante (E): **comportamento più furbo**

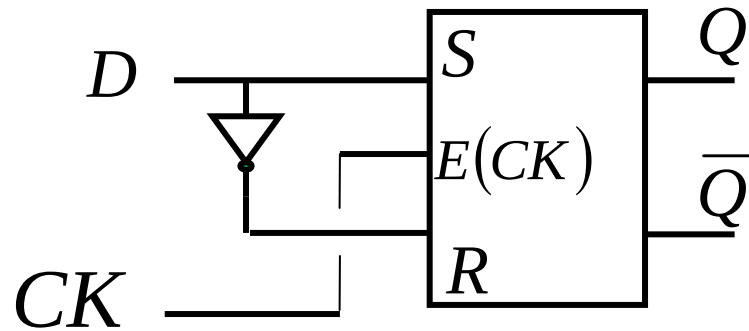
$$E=CK$$

In sostanza si hanno due modi:

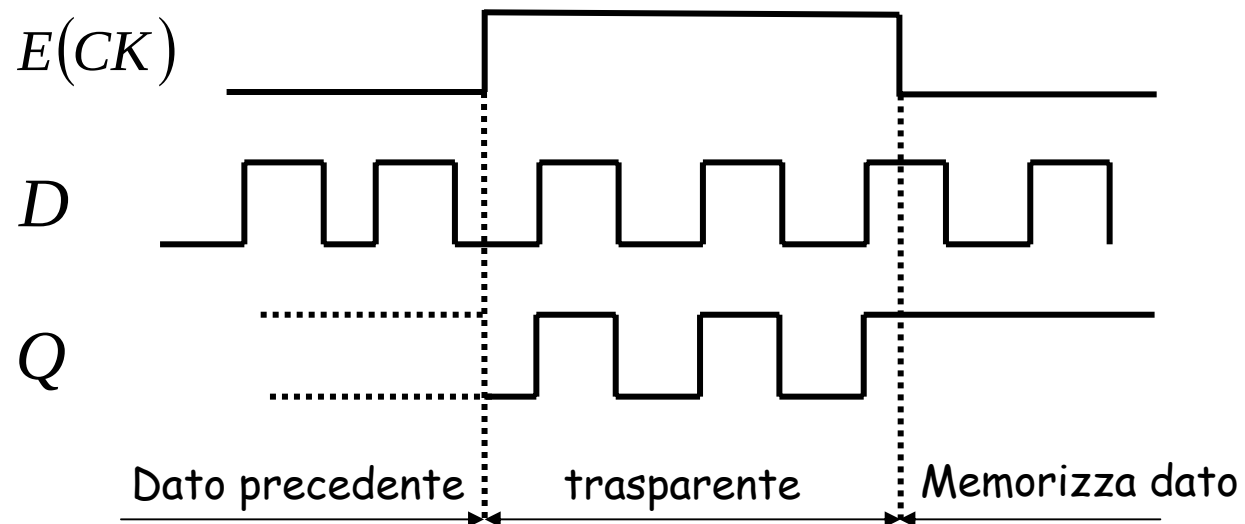
1. Abilitare il FF: E fisso mentre variano S e R
2. Ritardare opportunamente il FF: R e S sono stabili prima e durante un impulso (ciclo ?) di clock (E)

Il FF è una cella di memoria a un bit. Es: 7475 , 7477.....

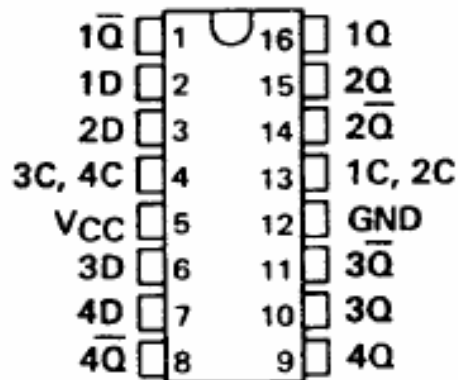
LATCH "trasparente"



E	D	Q_{n+1}
0	X	Q_n
1	D	D



SN5475, SN54LS75 . . . J OR W PACKAGE
 SN7475 . . . N PACKAGE
 SN74LS75 . . . D OR N PACKAGE
 (TOP VIEW)



FUNCTION TABLE
 (each latch)

INPUTS		OUTPUTS	
D	C	Q	$\bar{Q}$
L	H	L	H
H	H	H	L
X	L	Q_0	$\bar{Q}_0$

H = high level, L = low level, X = irrelevant

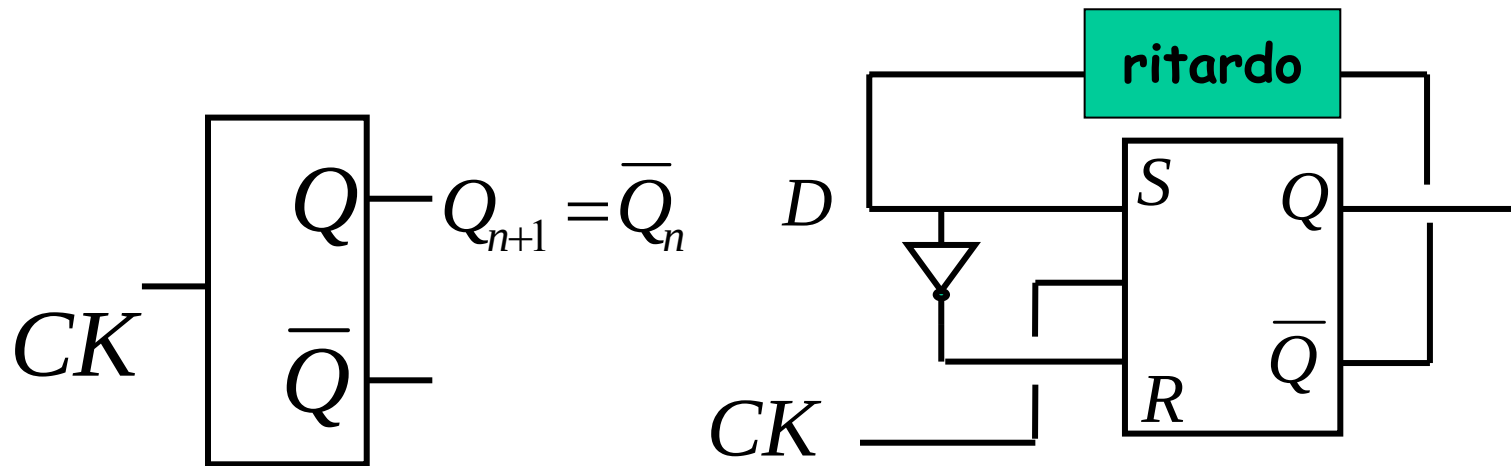
Q_0 = the level of Q before the high-to-low transition of G

Esistono altri tipi di FF:

1. Toggle: usato nei contatori, frequenzimetri, divisori ecc. ecc.

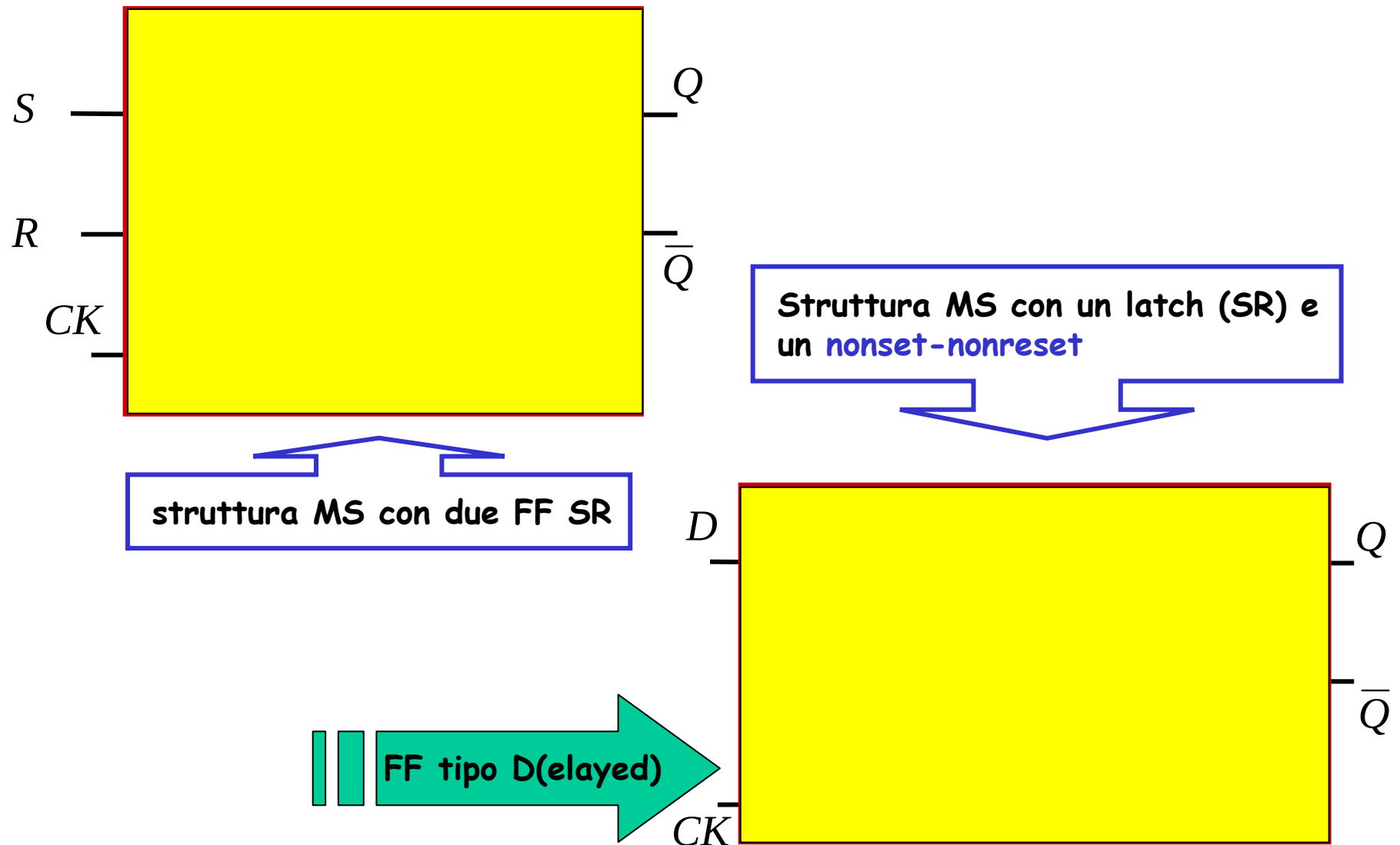
- Cambia stato in uscita ad ogni impulso di clock;
- **Attenzione**: non deve essere trasparente!!

Proviamo infatti a modificare il nostro LATCH



Ma non funziona!!!
A causa della trasparenza

Tipica struttura non trasparente: *Master-Slave*



Esperienza D-3

Comprende le prove:

- Circuiti Bistabili: Flip-Flop SR
- F-F SR con Enable (latch o memoria trasparente)
- Struttura Master-Slave come memoria non trasparente (D-type FF)
- Struttura Master-Slave come Toggle

Finalità della prova:

- Comportamento bi-stabile
- Differenza con i circuiti combinatori
- Ingressi attivi alti e attivi bassi
- Uso del FF come memoria ad 1 bit
- Trasparenza e non trasparenza

Realizzazione Pratica:

1. Realizzare i FF SR usando

- 2 porte NOR
- 2 porte NAND

Verificarne il funzionamento, senza smontarli alla fine, usando gli interruttori ed i led disponibili sulla basetta.

Stabilire un criterio per identificare:

- $Q, \overline{Q}, S, R$
- Scrivere le tabelline della verità per ciascun circuito

2. Cercate di rispondere ai seguenti quesiti:

- All' accensione, con S, R , non attivati, è prevedibile lo stato assunto dal FF?
- Se si toglie l' alimentazione (alla basetta) e poi si ripristina più volte, cosa si nota? Se lo stato assunto è sempre lo stesso può definirsi prevedibile?
- Cosa si osserva se si ripetono le prove precedenti con un solo dei due ingressi attivato? E con 2?
- Che significato ha ciascuna riga della tavola della verità rilevata?
- Verificare l' effetto della simultanea attivazione e disattivazione di S, R . (collegandoli ad un unico switch logico)

3. Realizzare i FF SR con Enable (completando i circuiti precedenti) usando:

- **4 porte NOR**
- **4 porte NAND**

Verificarne il funzionamento, senza smontarli alla fine, usando gli interruttori ed i led disponibili sulla basetta.

Stabilire un criterio per identificare:

- **$Q, \overline{Q}, S, R$**
- **Scivere le tabelline della verità per ciascun circuito**

4. Ripetere le prove eseguite con i FF con due sole porte

5. Verificare la differenza di funzionamento tra abilitazione e sincronizzazione

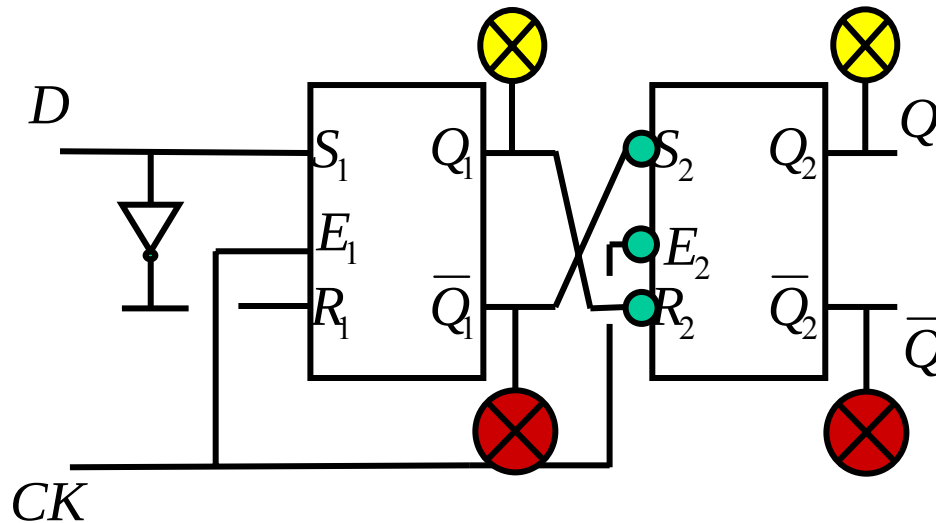
6. Rispondete alle seguenti domande:

- **Cosa significa ingressi attivi alti e attivi bassi?**
- **Che tipo di segnale si suppone di applicare all' ingresso del clock?**
- **Cosa succede attivando simultaneamente S,R?**
- **Cosa significa che le uscite del FF assumano eventualmente lo stesso stato?**

7. Modificare i circuiti in LATCH ponendo un inverter tra i due ingressi ed usando come clock un interruttore. Verificare la trasparenza facendo anche uso del F.G. E del C.R.O. Provare la possibilità di ottenere un Toggle mediante retroazione Ingresso-Uscita!!!!!!

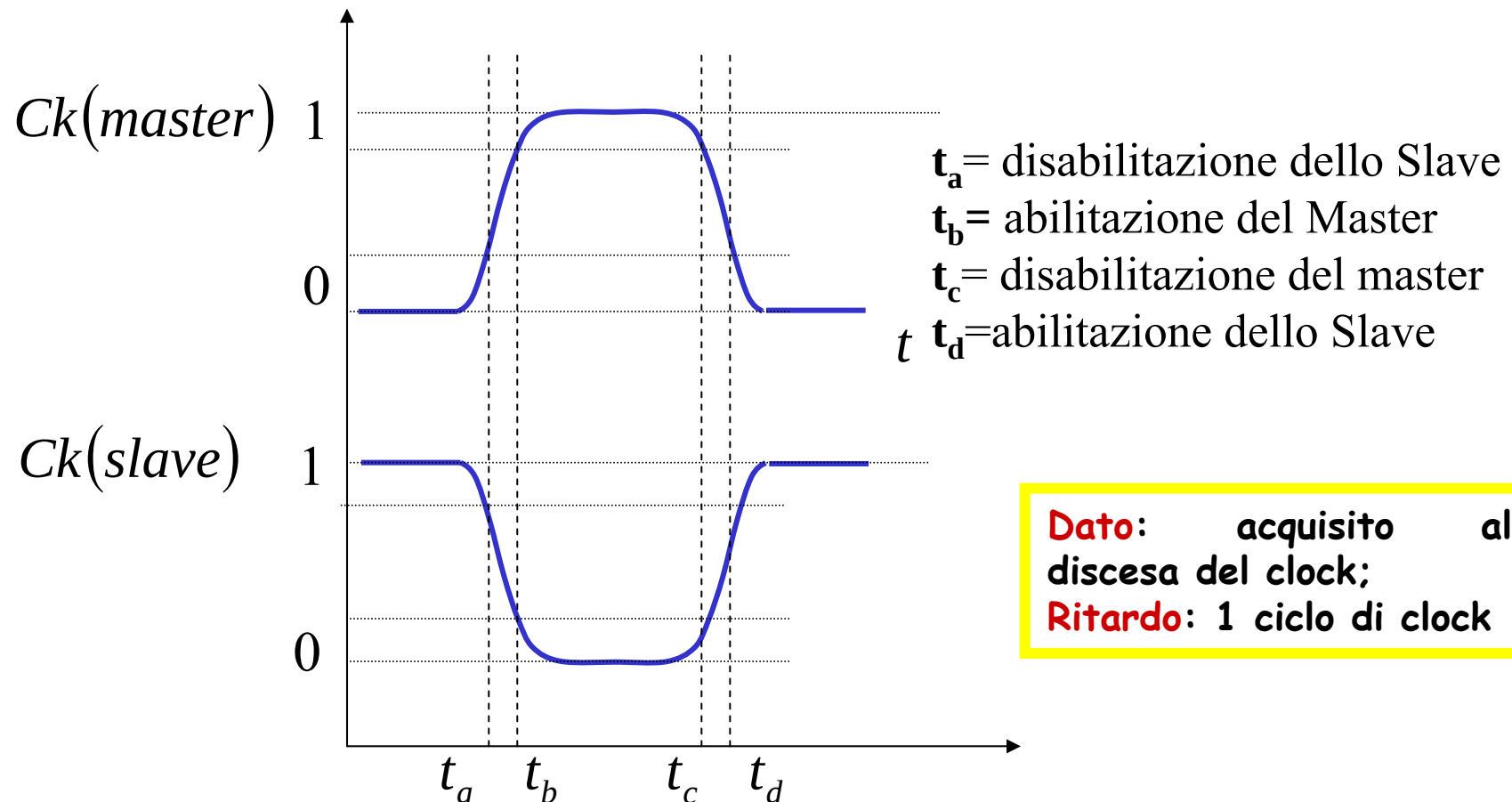
8. Struttura M-S: usare i due FF montati nelle prove precedenti

- D-type FF
- Toggle FF
- Debounced switch



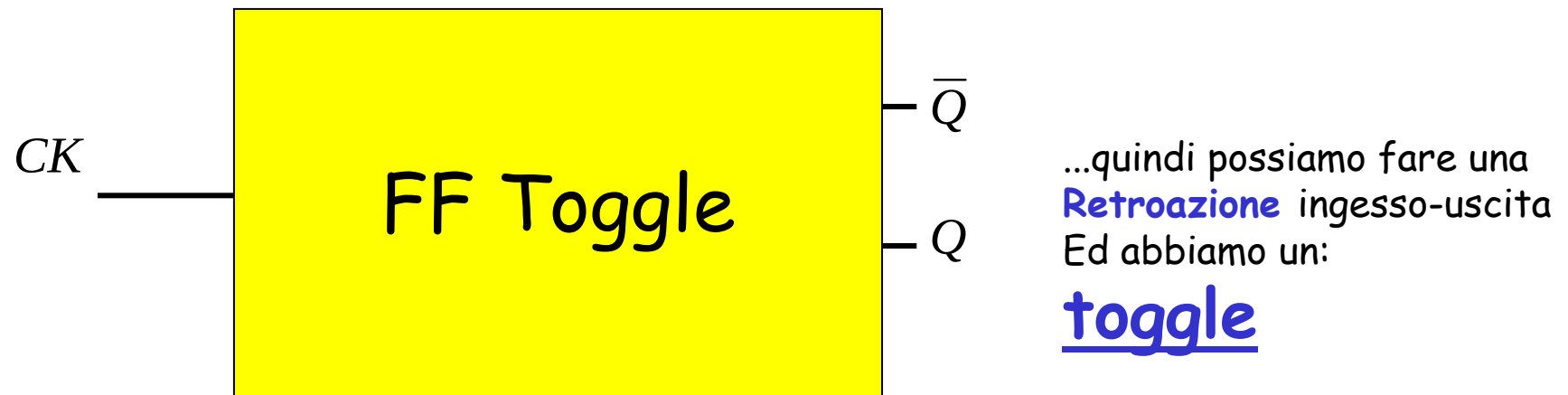
- Collegare le uscite ai led
- Collegare il Ck e D a due switch
- Verificare il funzionamento ed il Corretto trasferimento dei bit tra ingresso ed uscita
- in caso di irregolarità di funzionamento.....
- Eliminando la linea dei dati, eseguire la retroazione ingresso-uscita e realizzare il toggle

Adesso non ho più il problema del tempo di propagazione se la struttura non è trasparente.....

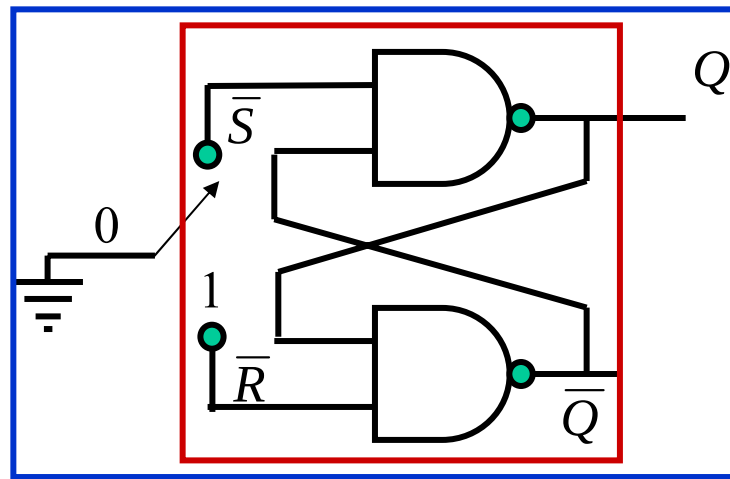


Dato: acquisito alla discesa del clock;
Ritardo: 1 ciclo di clock

Adesso non ho più il problema del tempo di propagazione se la struttura non è trasparente.....



Avevamo visto una semplice ma utilissima applicazione del FF?

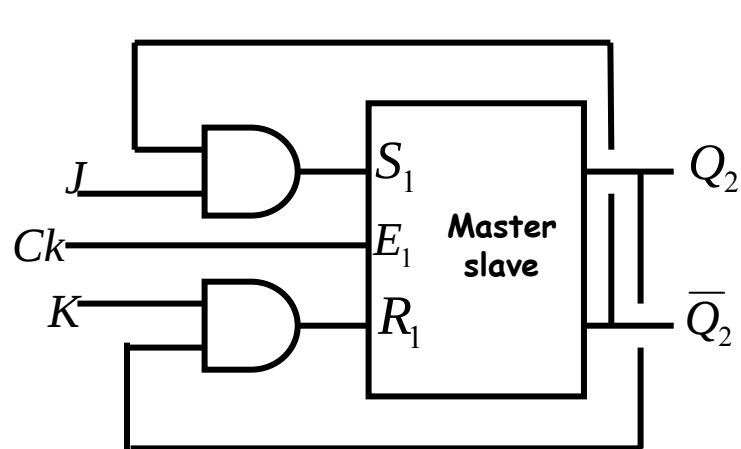


È anti-rimbalzo!

1. Il M.S. Sembra trasparente
2. Il toggle sembra non commutare

provare per credere!!!

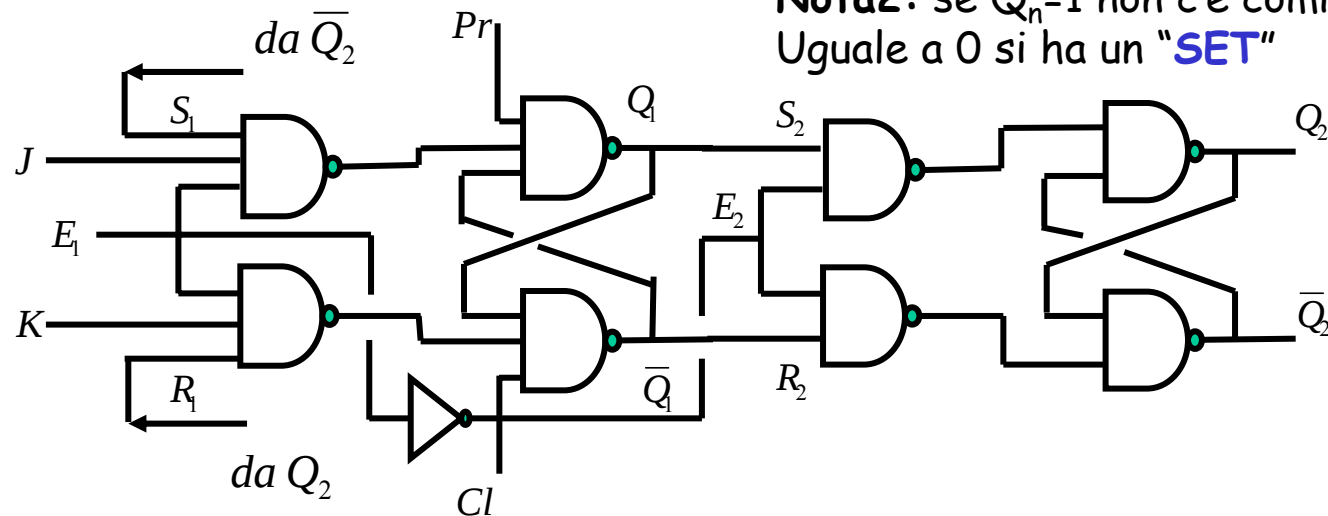
Ma si può fare di più.... Aggiungiamo alcuni "controlli"
 Aggiungiamo due **ENABLE GATES** e vediamo come funziona il tutto



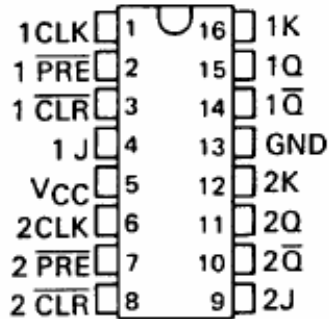
J	K	$S = J\overline{Q}_n$	$R = KQ_n$	Q_{n+1}
0	0	0	0	Q_n
0	1	0	Q_n	0 <i>nota 1</i>
1	0	$\overline{Q}_n$	0	1 <i>nota 2</i>
1	1	$\overline{Q}_n$	Q_n	$Q_n:Toggle$

Nota1: se $Q_n=0$ non c'è commutazione; se è Uguale a 1 si ha un "RESET"

Nota2: se $Q_n=1$ non c'è commutazione; se è Uguale a 0 si ha un "SET"



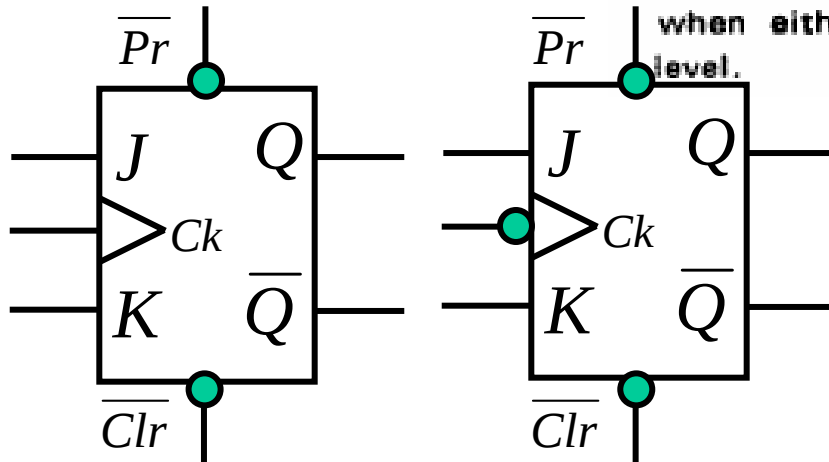
SN5476, SN54LS76A . . . J PACKAGE
 SN7476 . . . N PACKAGE
 SN74LS76A . . . D OR N PACKAGE
 (TOP VIEW)



'76
 FUNCTION TABLE

INPUTS					OUTPUTS	
PRE	CLR	CLK	J	K	Q	$\bar{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H [†]	H [†]
H	H	⌋	L	L	Q ₀	$\bar{Q}_0$
H	H	⌋	H	L	H	L
H	H	⌋	L	H	L	H
H	H	⌋	H	H	TOGGLE	

[†] This configuration is nonstable; that is, it will not persist when either preset or clear returns to its inactive (high) level.



J	K	Q _{n+1}
0	0	Q _n
0	1	1
1	0	0
1	1	Toggle

↓ L' uscita cambia alla discesa del clock

↑ L' uscita cambia alla salita del clock



L' uscita cambia un impulso completo del clock. Gli ingressi devono essere stabili prima e durante l' impulso. E la transizione si ha alla fine.

Riassumiamo la situazione:

I FF di tipo JK

1. Hanno tre ingressi principali: **J, K, Ck**;
2. Hanno due ingressi accessori: **Pr, Clr**
3. Pr, Clr possono essere: **sincroni, asincroni**
4. Funzionano come **SRFF** (Clocked), **Toggle**
5. Sono "non trasparenti" in quanto EDGE triggered
6. J, K, Ck possono essere: **attivi alti/ attivi bassi**
7. Il dato all' uscita appare sempre alla fine del ciclo di Clock ma:
 - Struttura **MS**: i dati devono essere stabili prima e durante l' impulso di Clock
 - Struttura **true edge triggered**: i dati devono essere stabili solo sul fronte di commutazione

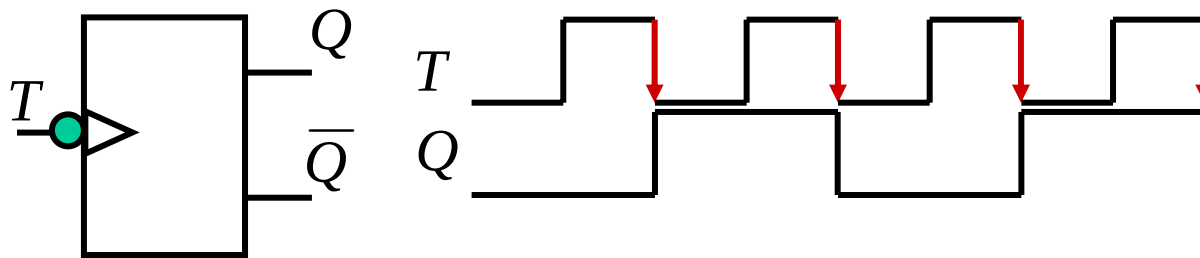
Definizioni generali importanti:

T_{su} =**setup time**: tempo minimo tra stabilizzazione degli ingressi ed attivazione del Clock

T_h =**hold time**: tempo minimo durante il quale gli ingressi devono essere stabili dopo la transizione del Ck

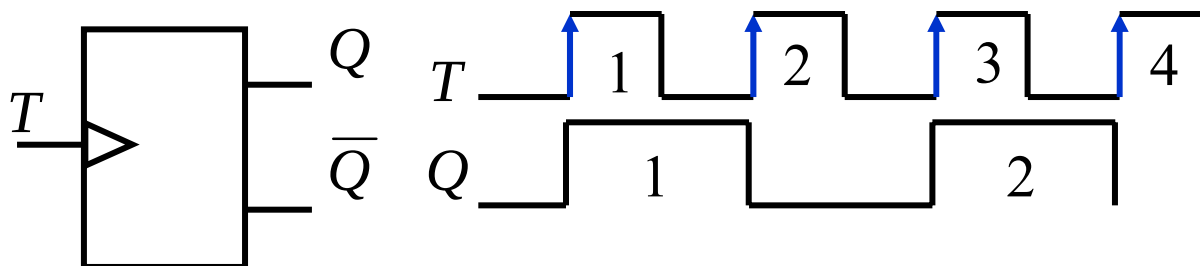
T_w =**time width**: tempo minimo della durata del Ck per avere un funzionamento corretto

Abbiamo realizzato un **FF Toggle** basandoci sulla retroazione
ingresso-uscita

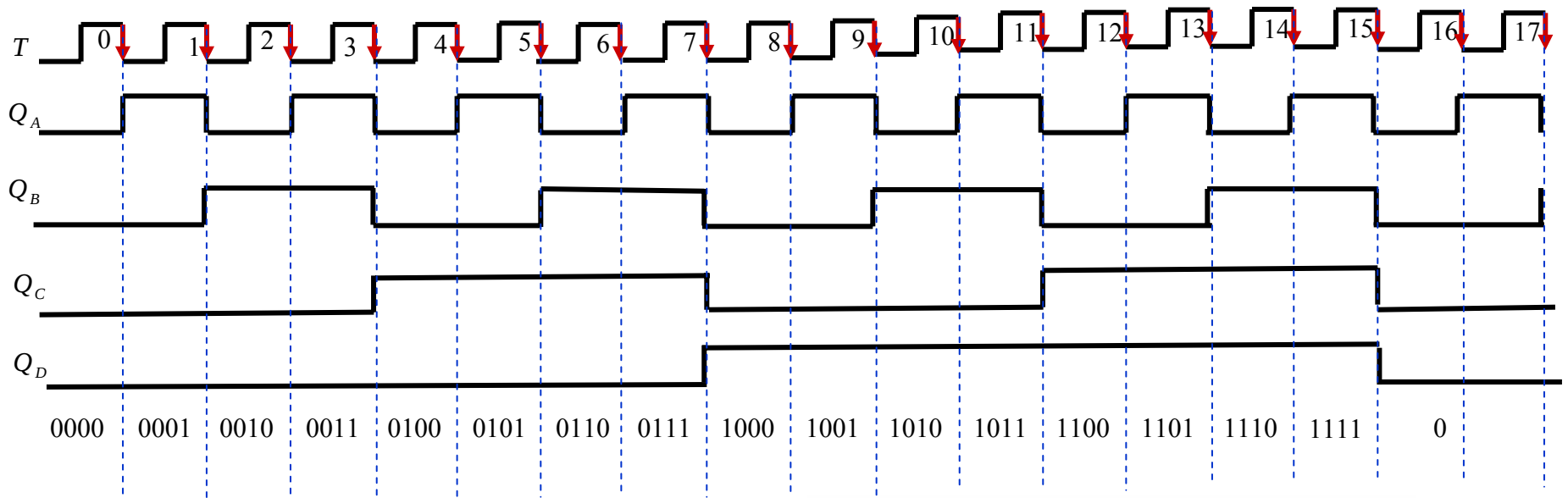


T	Q_n	Q_{n+1}
0	0	0
1	1	1
↓	0	1
↓	1	0

E' un divisore per due che commuta sul fronte:



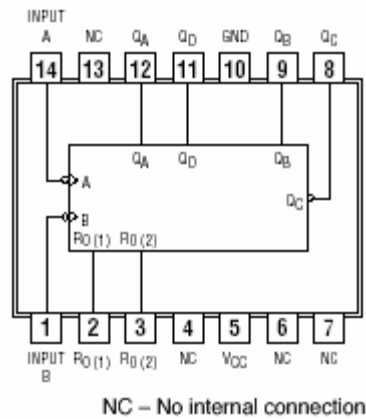
T	Q_n	Q_{n+1}
0	0	0
1	1	1
↑	0	1
↑	1	0



TRUTH TABLE

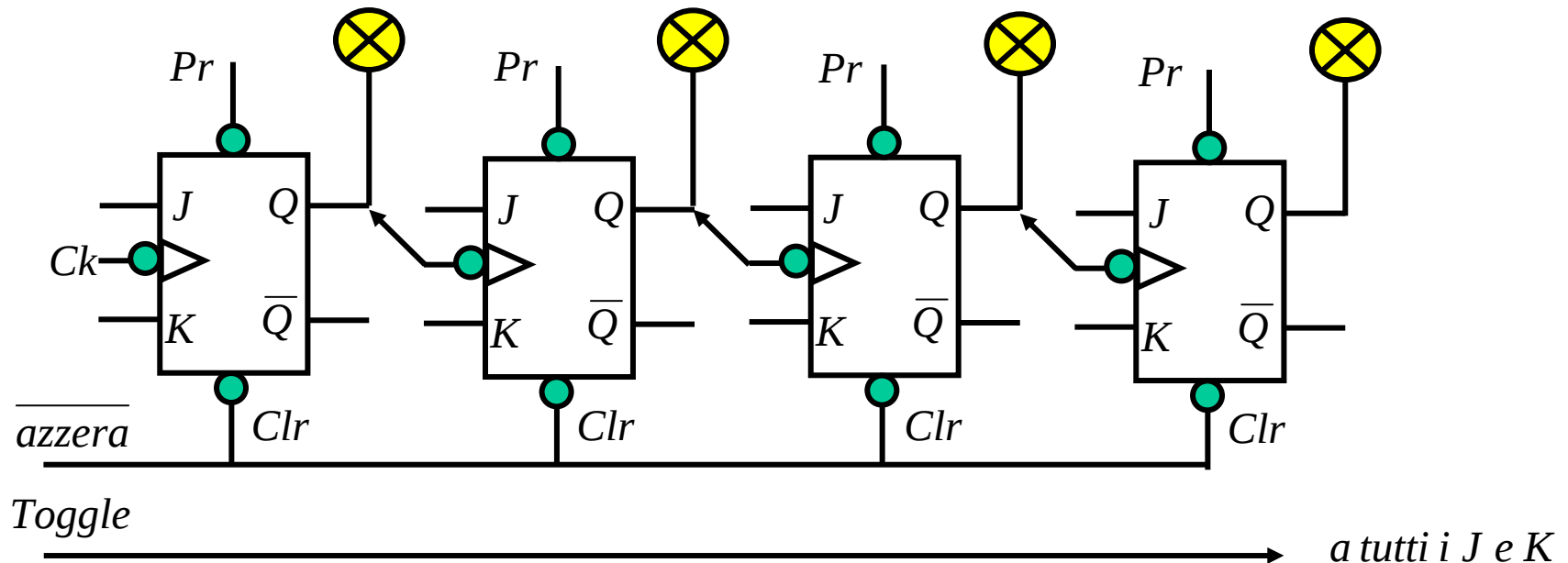
COUNT	OUTPUTS			
	Q_0	Q_1	Q_2	Q_3
0	L	L	L	L
1	H	L	L	L
2	L	H	L	L
3	H	H	L	L
4	L	L	H	L
5	H	L	H	L
6	L	H	H	L
7	H	H	H	L
8	L	L	L	H
9	H	L	L	H
10	L	H	L	H
11	H	H	L	H
12	L	L	H	H
13	H	L	H	H
14	L	H	H	H
15	H	H	H	H

H = High Voltage Level, L = Low Voltage Level



NC – No internal connection

Contatore a 4 bit, asincrono, con linee di azzeramento e toggle (blocca se basso Conta se alto). Il conteggio avviene avanti o indietro



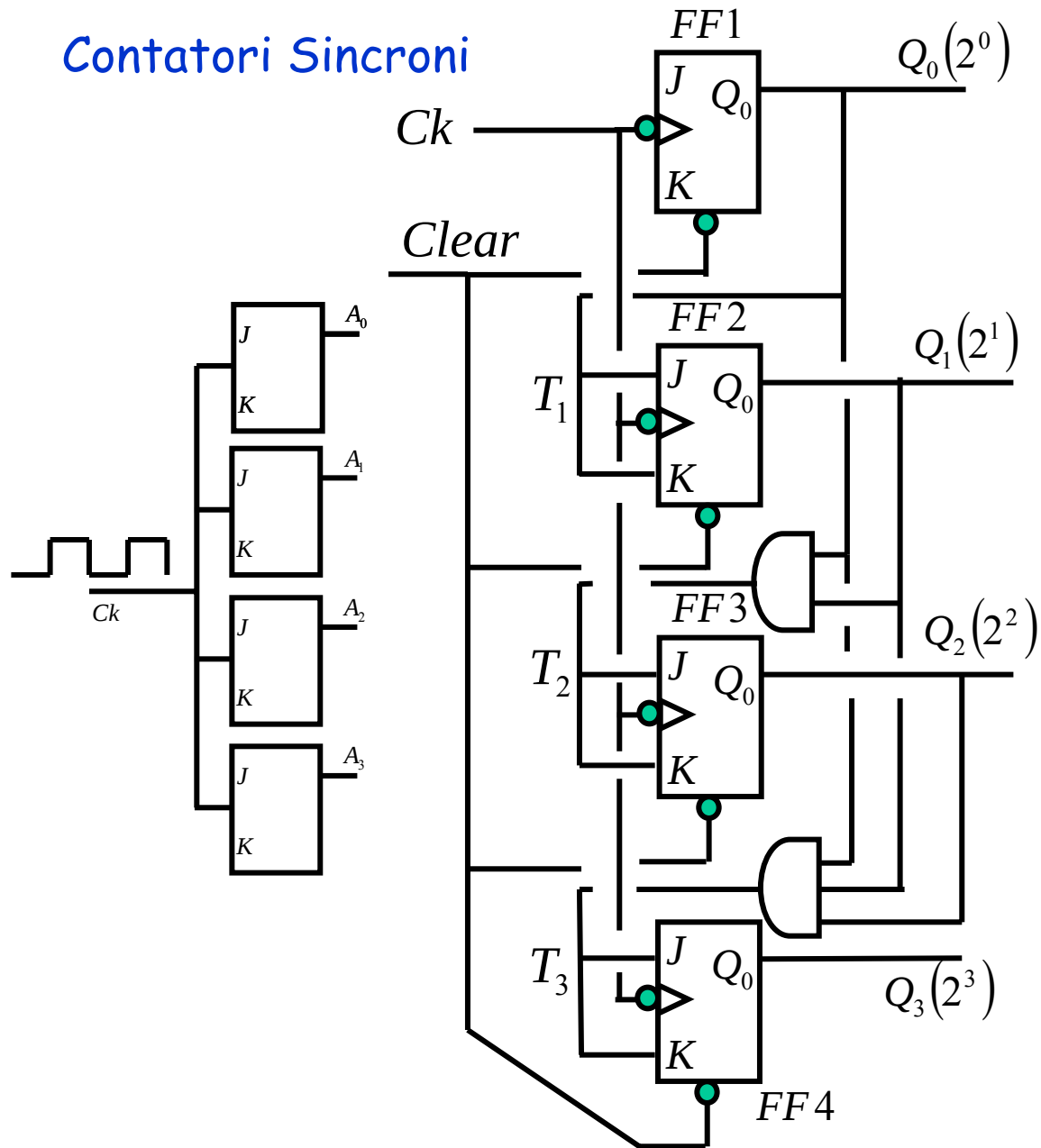
I contatori di questo tipo sono detti : **RIPPLE COUNTERS**

Problema: e se cambiano contemporaneamente più uscite?

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



Contatori Sincroni



TRUTH TABLE

COUNT	OUTPUTS			
	Q_0	Q_1	Q_2	Q_3
0	L	L	L	L
1	H	L	L	L
2	L	H	L	L
3	H	H	L	L
4	L	L	H	L
5	H	L	H	L
6	L	H	H	L
7	H	H	H	L
8	L	L	L	H
9	H	L	L	H
10	L	H	L	H
11	H	H	L	H
12	L	L	H	H
13	H	L	H	H
14	L	H	H	H
15	H	H	H	H

H = High Voltage Level, L = Low Voltage Level

'160, '161, 'LS160A, 'LS161A . . . SYNCHRONOUS COUNTERS WITH DIRECT CLEAR
'162, '163, 'LS162A, 'LS163A, 'S162, 'S163 . . . FULLY SYNCHRONOUS COUNTERS

- Internal Look-Ahead for Fast Counting
- Carry Output for n-Bit Cascading
- Synchronous Counting
- Synchronously Programmable
- Load Control Line
- Diode-Clamped Inputs

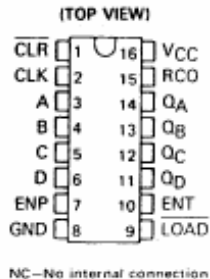
TYPE	TYPICAL PROPAGATION TIME, CLOCK TO Q OUTPUT	TYPICAL MAXIMUM CLOCK FREQUENCY	TYPICAL POWER DISSIPATION
'160 thru '163	14 ns	32 MHz	305 mW
'LS162A thru 'LS163A	14 ns	32 MHz	93 mW
'S162 and 'S163	9 ns	70 MHz	475 mW

description

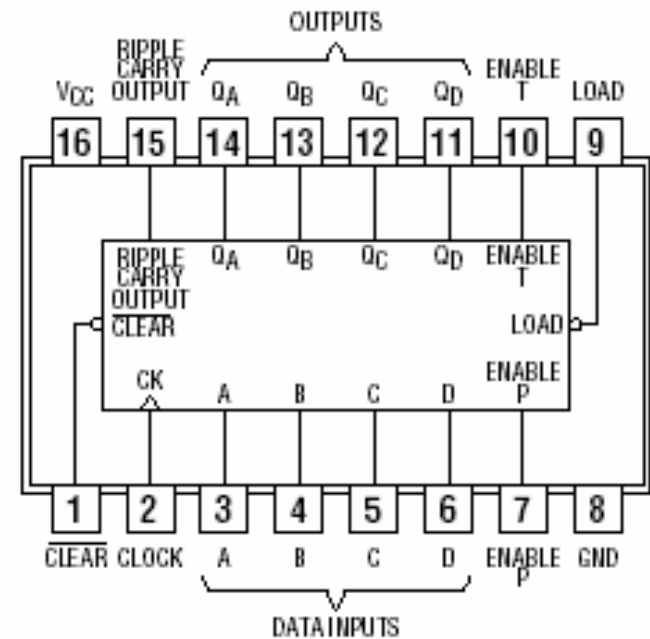
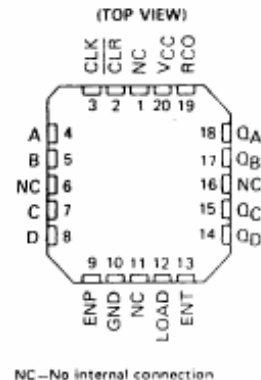
These synchronous, presettable counters feature an internal carry look-ahead for application in high-speed counting designs. The '160, '162, 'LS160A, 'LS162A, and 'S162 are decade counters and the '161, '163, 'LS161A, 'LS163A, and 'S163 are 4-bit binary counters. Synchronous operation is provided by having all flip-flops clocked simultaneously so that the outputs change coincident with each other when so instructed by the count-enable inputs and internal gating. This mode of operation eliminates the output counting spikes that are normally associated with asynchronous (ripple clock) counters, however counting spikes may occur on the (RCO) ripple carry output. A buffered clock input triggers the four flip-flops on the rising edge of the clock input waveform.

These counters are fully programmable; that is, the outputs may be preset to either level. As presetting is synchronous, setting up a low level at the load input disables the counter and causes the outputs to agree with the setup data after the next clock pulse regardless of the levels of the enable inputs. Low-to-high transitions at the load input of the '160 thru '163 should be avoided when the clock is low if the enable inputs are high at or before the transition. This restriction is not applicable to the 'LS160A thru 'LS163A or 'S162 or 'S163. The clear function for the '160, '161, 'LS160A, and 'LS161A is asynchronous and a low level at the clear input sets all four of the flip-flop outputs low regardless of the levels of clock, load, or enable inputs. The clear function for the '162, '163, 'LS162A, 'LS163A, 'S162, and 'S163 is synchronous and a low level at the clear input sets all four of the flip-flop outputs low after the next clock pulse, regardless of the levels of the enable inputs. This synchronous clear allows the count length to be modified easily as decoding the maximum count desired can be accomplished with one external NAND gate. The gate output is connected to the clear input to synchronously clear the counter to 0000 (LLLL). Low-to-high transitions at the clear input of the '162 and '163 should be avoided when the clock is low if the enable and load inputs are high at or before the transition.

SERIES 54', 54LS', 54S' . . . J OR W PACKAGE
SERIES 74' . . . N PACKAGE
SERIES 74LS', 74S' . . . D OR N PACKAGE

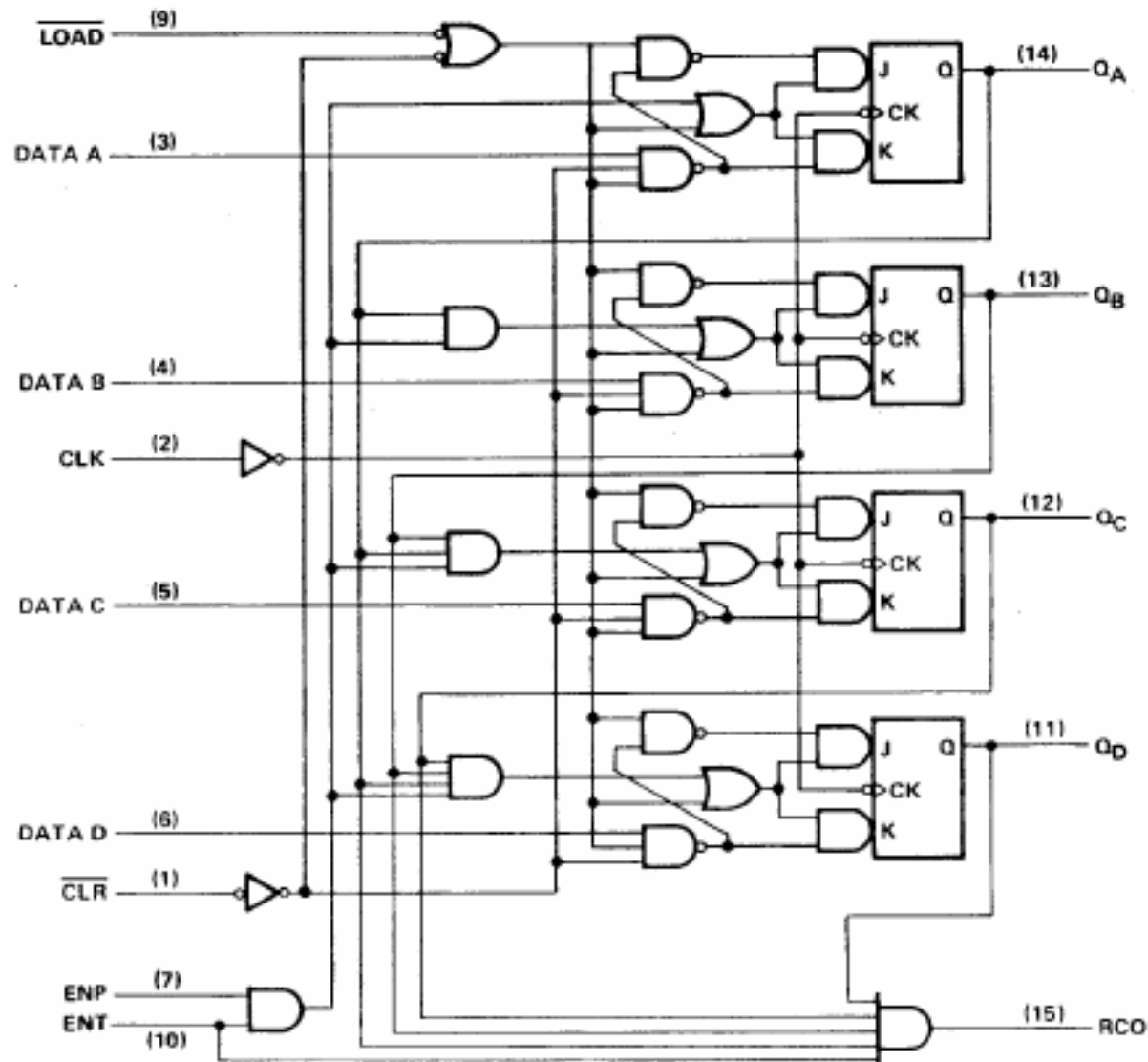


SERIES 54LS', 54S' . . . FK PACKAGE



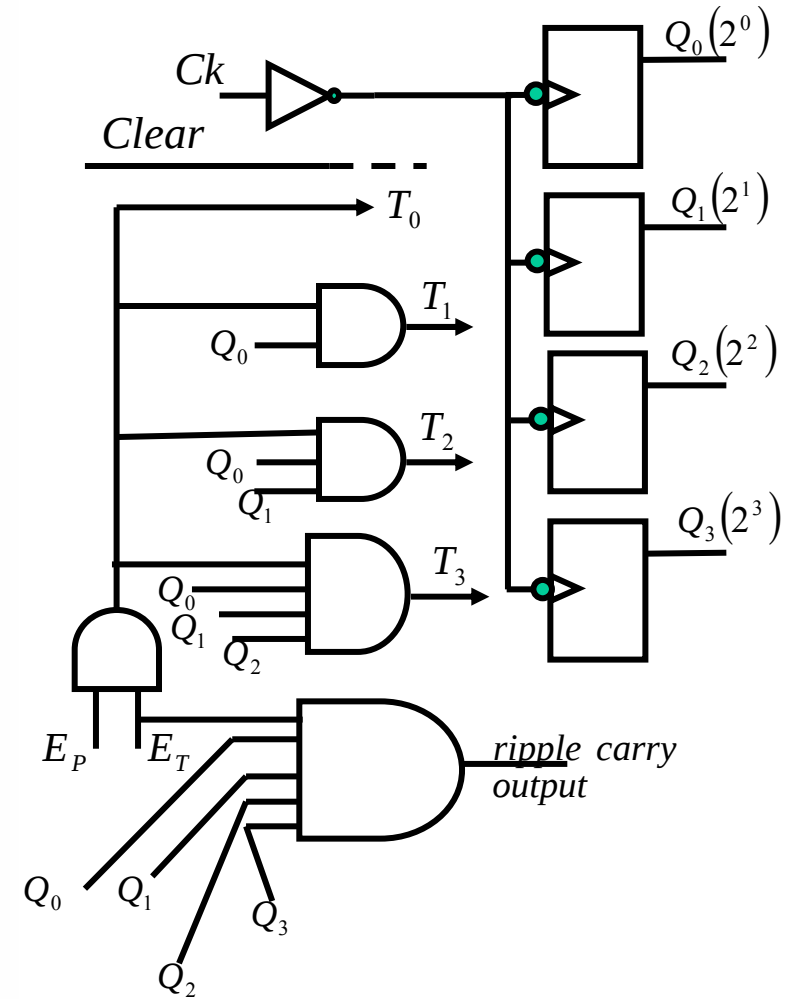
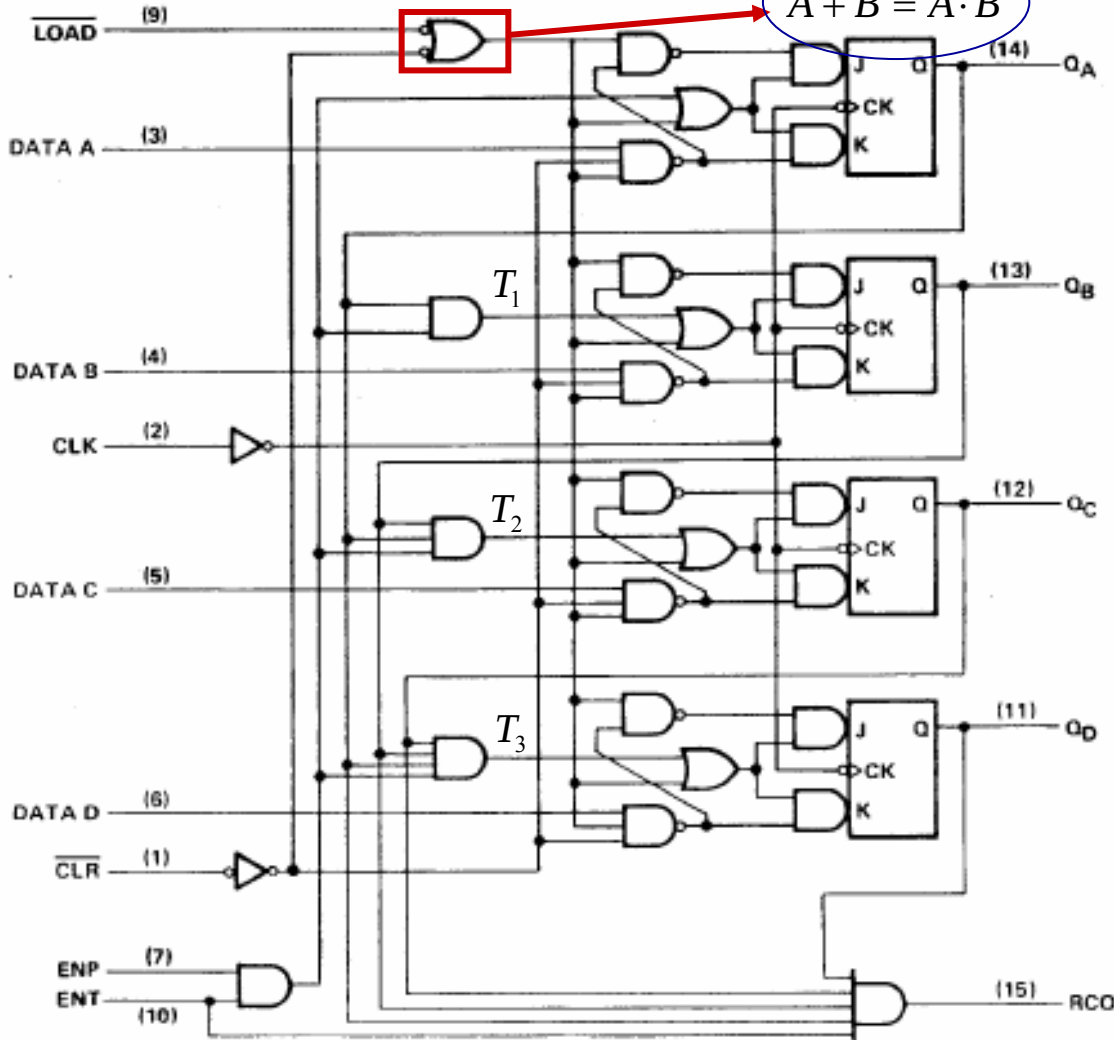
SN54163, SN74163 SYNCHRONOUS BINARY COUNTERS

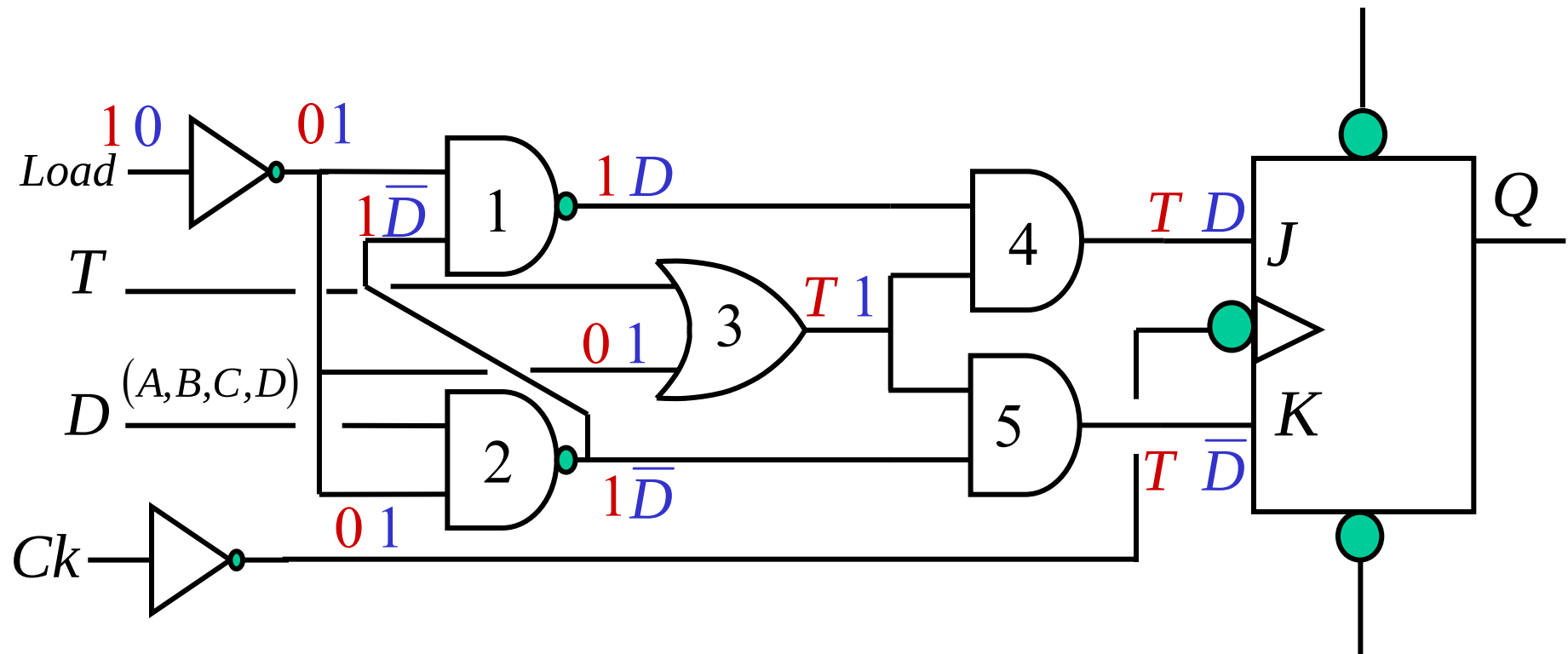
SN54161, SN74161 synchronous binary counters are similar; however, the clear is asynchronous as shown for the SN54160, SN74160 decade counters at left.



SN54163, SN74163 SYNCHRONOUS BINARY COUNTERS

Enable the complement!



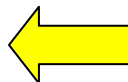


T: funzione di abilitazione vista prima

1. **Load=1**: T arriva a J e K $\rightarrow$ contatore normale

2. **Load=0**: D arriva a J, $\bar{D}$ arriva a K $\rightarrow$ **D type FF**

"Preset" del contatore



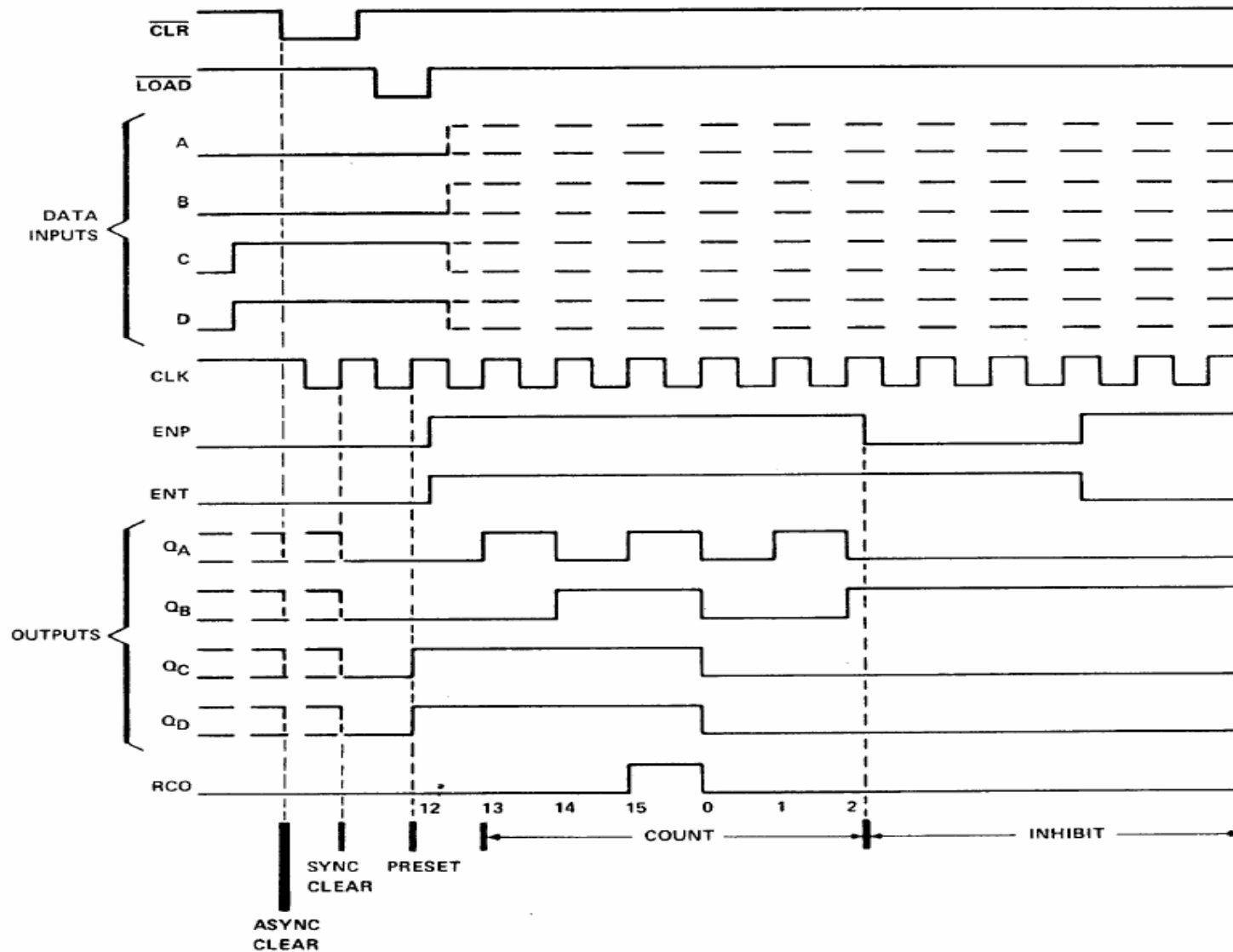
Q= D quando il Ck è attivato

'161, 'LS161A, '163, 'LS163A, 'S163 BINARY COUNTERS

typical clear, preset, count, and inhibit sequences

Illustrated below is the following sequence:

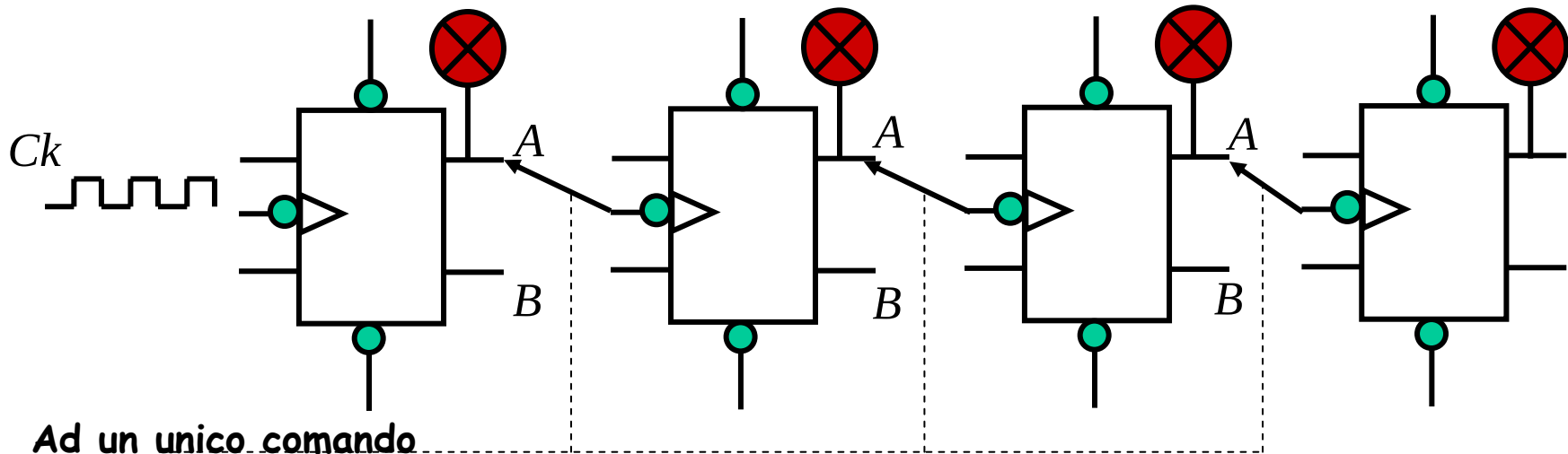
1. Clear outputs to zero ('161 and 'LS161A are asynchronous; '163, 'LS163A, and 'S163 are synchronous)
2. Preset to binary twelve
3. Count to thirteen, fourteen fifteen, zero, one, and two
4. Inhibit



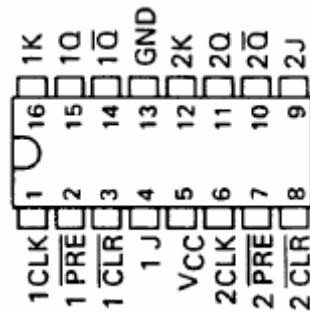
Esperienza D-4

Comprende le prove:

- Uso di un JK FF con Preset e Clear
- Contatore binario asincrono avanti/indietro a 4 bit con 4 FF JK
- Registro a scorrimento a 4 bit con 4 FF JK:
 - Funzionamento SISO, SIPO, PIPO, PISO.



SN5476, SN54LS76A . . . J PACKAGE
 SN7476 . . . N PACKAGE
 SN74LS76A . . . D OR N PACKAGE
 (TOP VIEW)

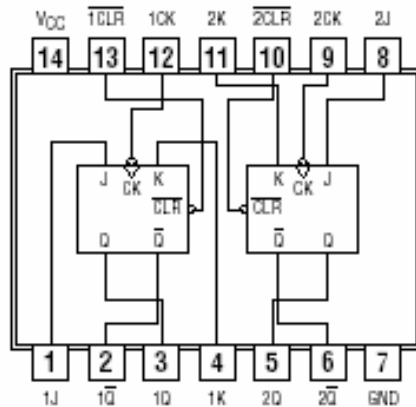


'76
 FUNCTION TABLE

INPUTS					OUTPUTS	
PRE	CLR	CLK	J	K	Q	Q̄
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H [†]	H [†]
H	H	⌋	L	L	Q ₀	Q̄ ₀
H	H	⌋	H	L	H	L
H	H	⌋	L	H	L	H
H	H	⌋	H	H	TOGGLE	

107

DUAL J-K FLIP-FLOPS WITH CLEAR



FUNCTION TABLES
 '107

INPUTS				OUTPUTS	
CLR	CLOCK	J	K	Q	Q̄
L	X	X	X	L	H
H	⌋	L	L	Q ₀	Q̄ ₀
H	⌋	H	L	H	L
H	⌋	L	H	L	H
H	⌋	H	H	TOGGLE	

'LS107A,'HC107

INPUTS				OUTPUTS	
CLR	CLOCK	J	K	Q	Q̄
L	X	X	X	L	H
H	↓	L	L	Q ₀	Q̄ ₀
H	↓	H	L	H	L
H	↓	L	H	L	H
H	↓	H	H	TOGGLE	
H	H	X	X	Q ₀	Q̄ ₀

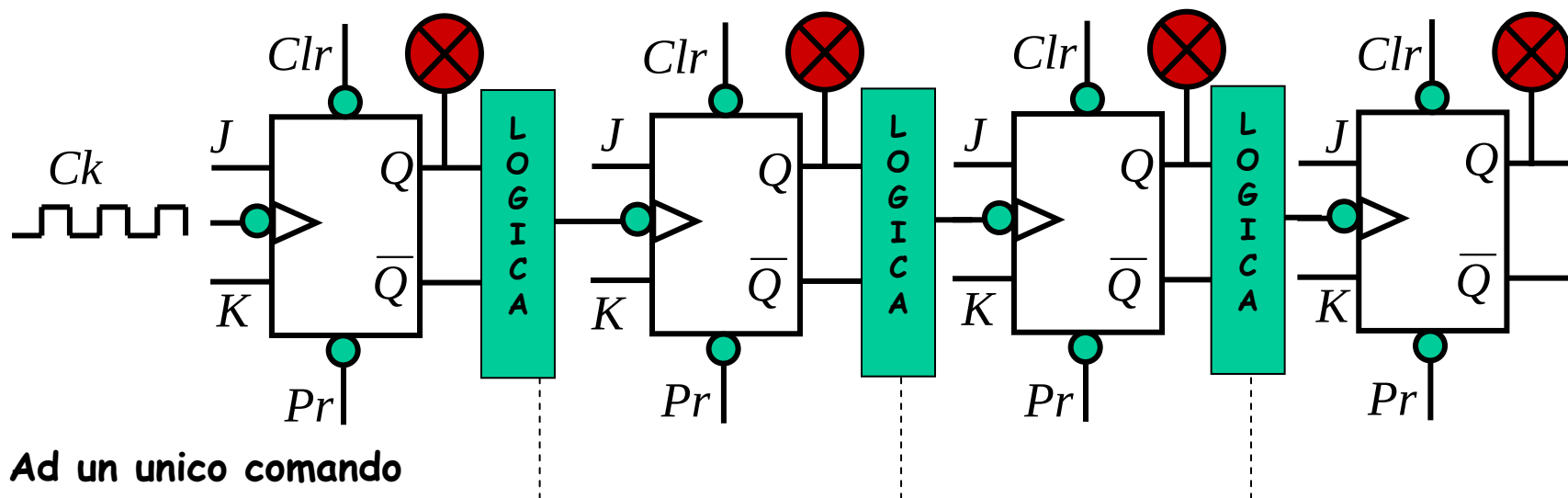
Selettori in posizione "A"

N	Q ₃	Q ₂	Q ₁	Q ₀	C
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	10
11	1	0	1	1	11
12	1	1	0	0	12
13	1	1	0	1	13
14	1	1	1	0	14
15	1	1	1	1	15
16	0	0	0	0	0

Selettori in posizione "B"

N	Q ₃	Q ₂	Q ₁	Q ₀	C
0	0	0	0	0	0
1	1	1	1	1	15
2	1	1	1	0	14
3	1	1	0	1	13
4	1	1	0	0	12
5	1	0	1	1	11
6	1	0	1	0	10
7	1	0	0	1	9
8	1	0	0	0	8
9	0	1	1	1	7
10	0	1	1	0	6
11	0	1	0	1	5
12	0	1	0	0	4
13	0	0	1	1	3
14	0	0	1	0	2
15	0	0	0	1	1
16	0	0	0	0	0

Ma come si passa da A a B in modo efficiente?

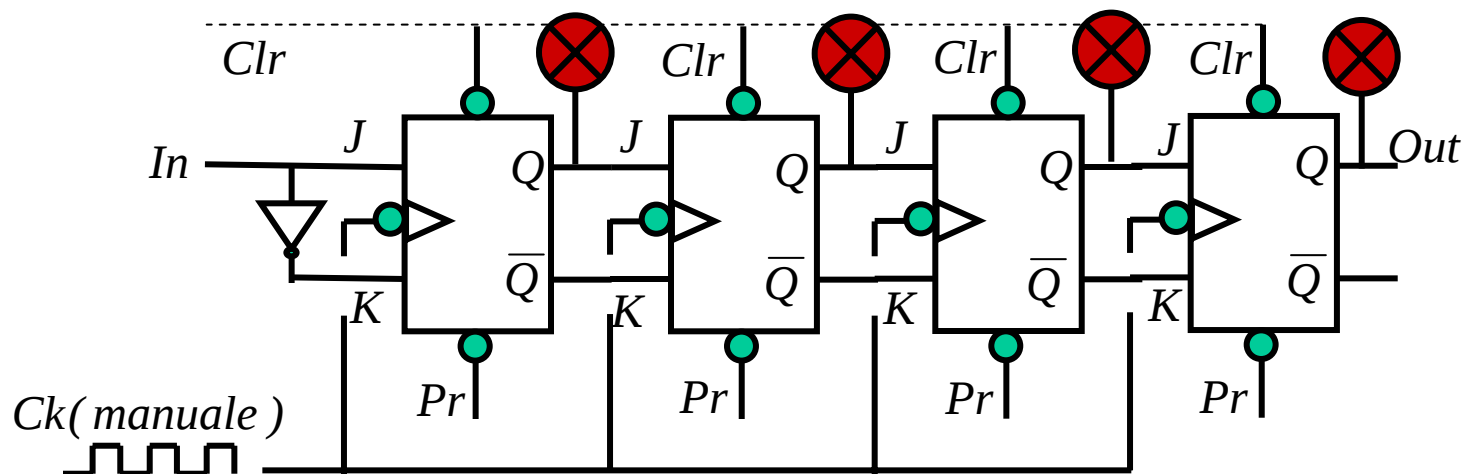


La logica può essere:
 1. MUX: 74157
 2. T/C: 7486

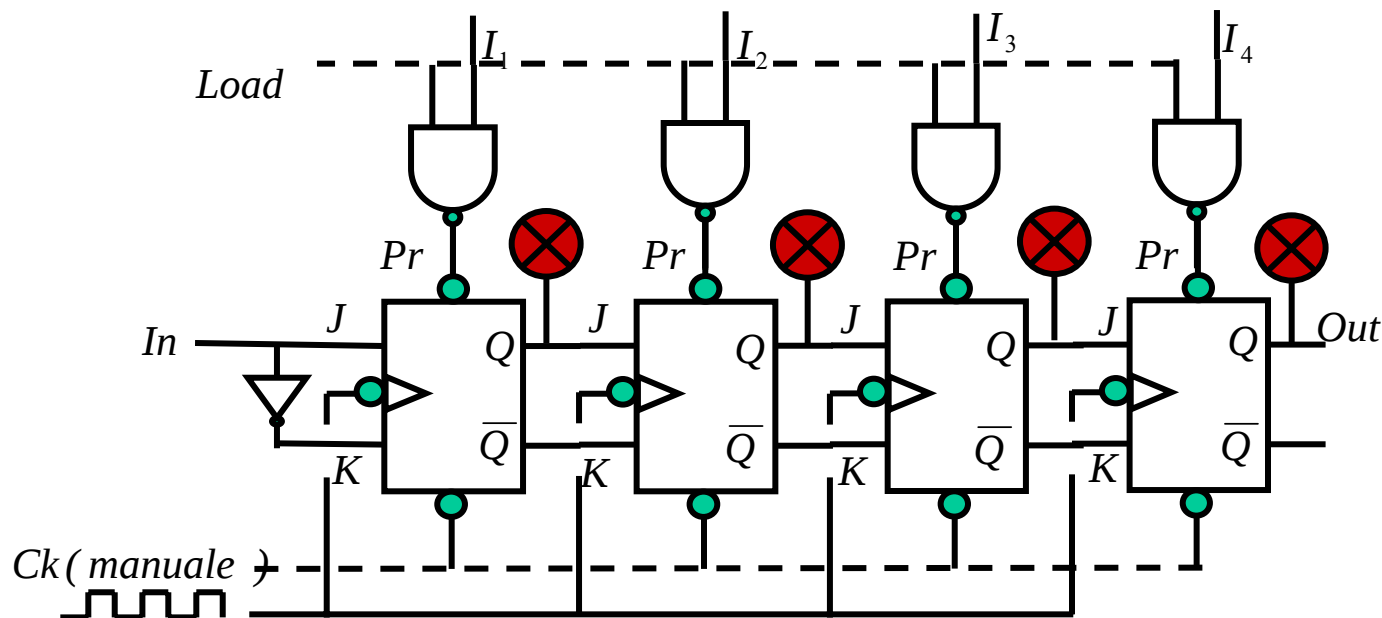
Provare il circuito usando:

- prima gli interruttori (per il Ck un anti-rimbalzo) e le uscite a 4 led
- poi usare il clock della basetta (o F.G.) ed il display a 7 segmenti.
 - Attivare Clr in varie condizioni dei restanti ingressi:cosa accade?
 - Attivare Clear e porre il Toggle a 1: che accade?
 - Azionare il selettore avanti/indietro:che succede?(occhio alle configurazioni!)
 - Mentre il contatore conta, portare il Toggle a zero: che succede?

Registro a scorrimento con 4 FF JK (7476): SISO,SIPO



Registro a scorrimento con 4 FF JK (7476): PISO,PIPO



Attenzione:

Non consente l'introduzione di zeri ma si può usare il clear.

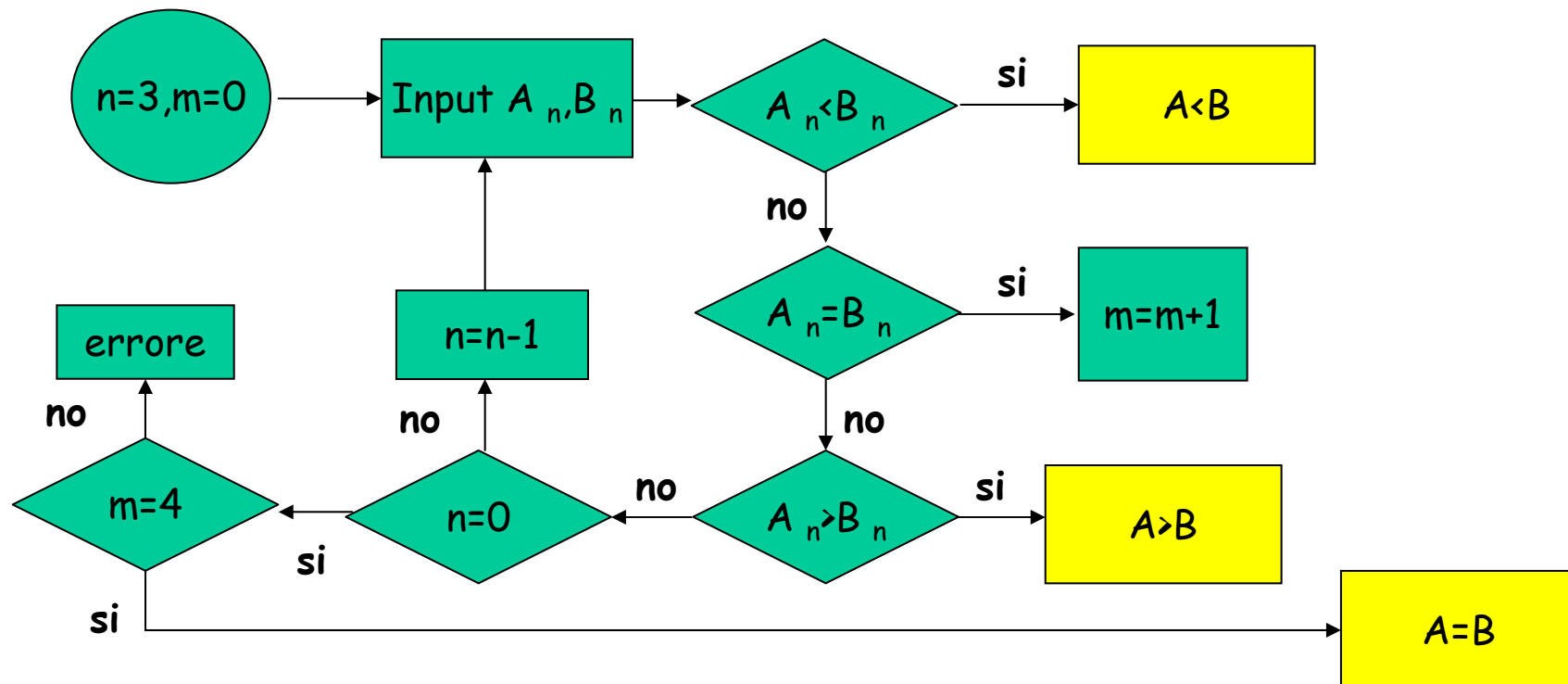
E' anche un:

Generatore di sequenza
Contatore ad anello

Comparatori

Abbiamo già visto il comparatore a un bit. Estendiamo il caso a 4 bit
Confrontare 4 bit significa:

1. Confrontare i bit più significativi
2. Decidere o..
3. Confrontare i bit di ordine immediatamente successivo
4. reiterare



Date due parole di 4 bit: $A_3, A_2, A_1, A_0, B_3, B_2, B_1, B_0$

E supponendo di avere le funzioni di eguaglianza (XNOR): E_3, E_2, E_1, E_0

$A_3 < B_3$

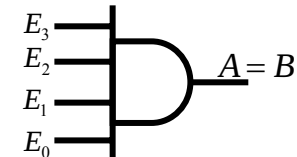
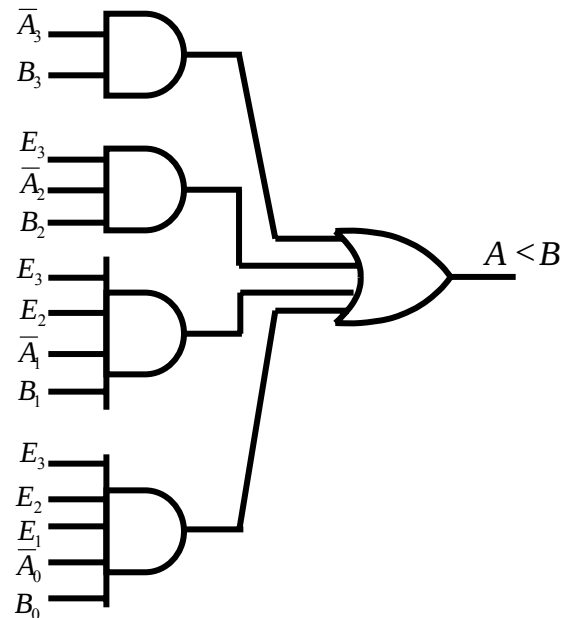
$A_2 < B_2$
 $A_3 = B_3$

$A_1 < B_1$
 $A_3 = B_3$
 $A_2 = B_2$

$A_0 < B_0$
 $A_3 = B_3$
 $A_2 = B_2$
 $A_1 = B_1$

B	A	$A \cdot B$
L	L	L
L	H	L
H	L	H
H	H	L

come le tratto queste condizioni?



N.B.

Ci servono solo:

- le funzioni di eguaglianza dei bit: E_n
- i complementi dei bit

85

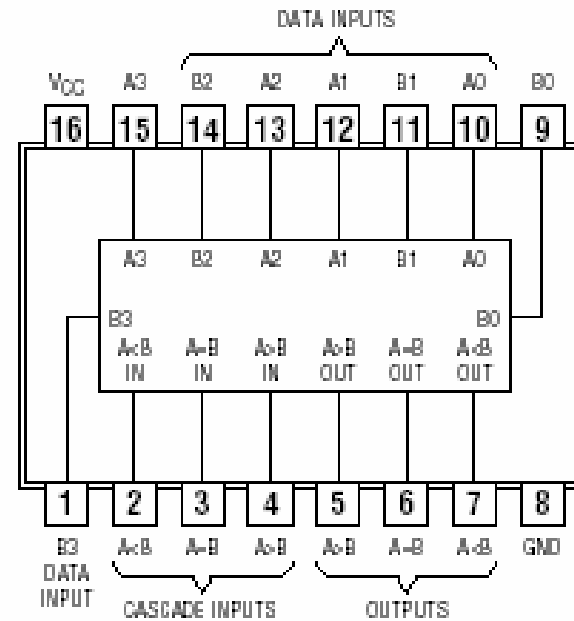
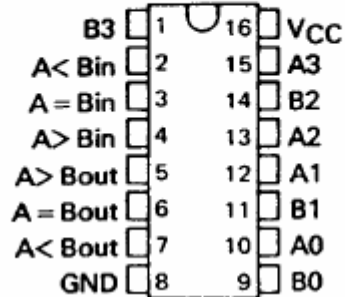
4-BIT MAGNITUDE COMPARATORS

SN5485, SN54LS85, SN54S85 . . . J OR W PACKAGE

SN7485 . . . N PACKAGE

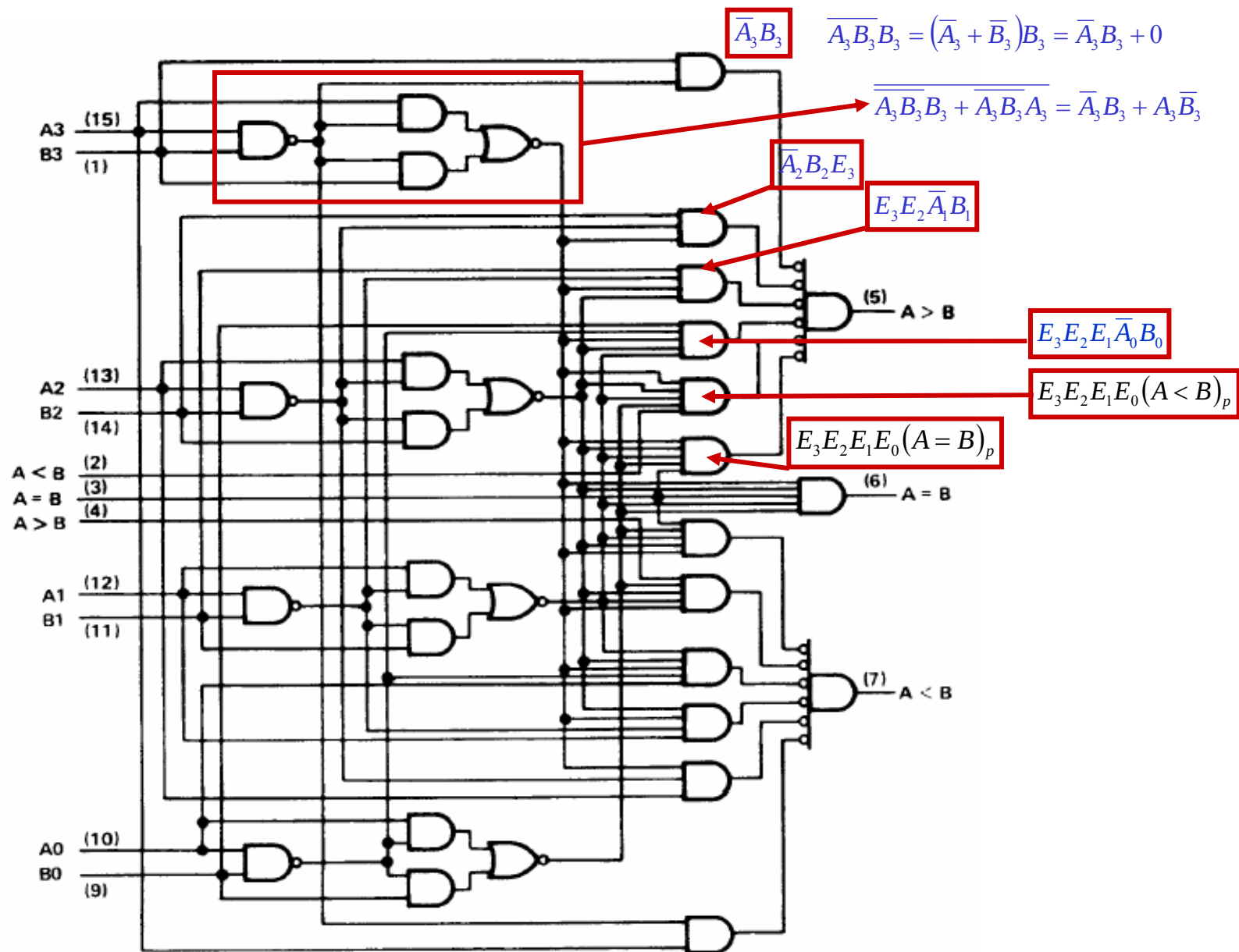
SN74LS85, SN74S85 . . . D OR N PACKAGE

(TOP VIEW)

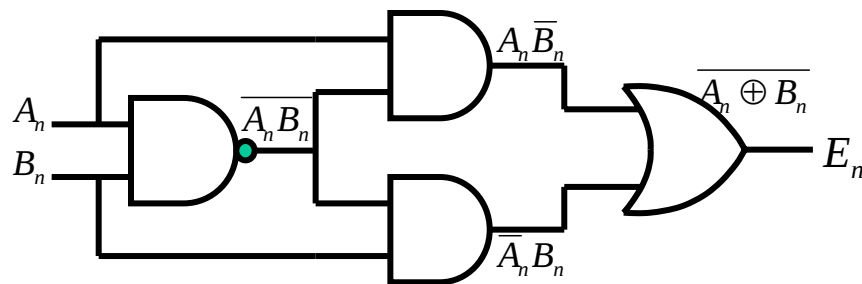


FUNCTION TABLE

COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A3, B3	A2, B2	A1, B1	A0, B0	A > B	A < B	A = B	A > B	A < B	A = B
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 = B3	A2 < B2	X	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

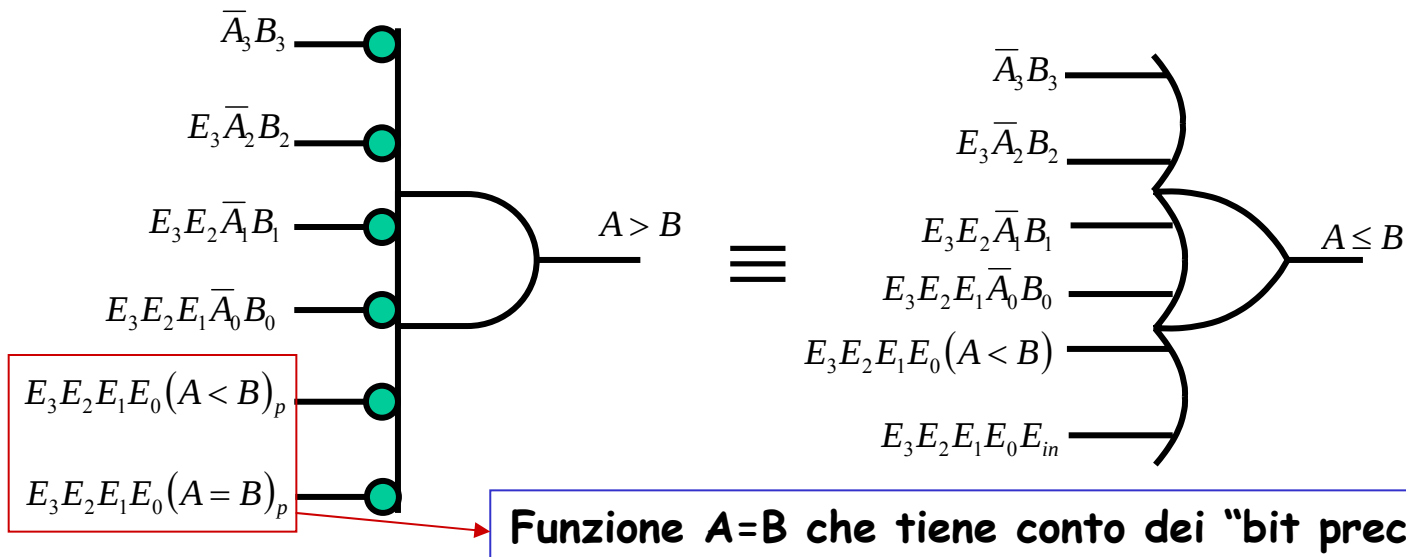


Le funzioni di eguaglianza:

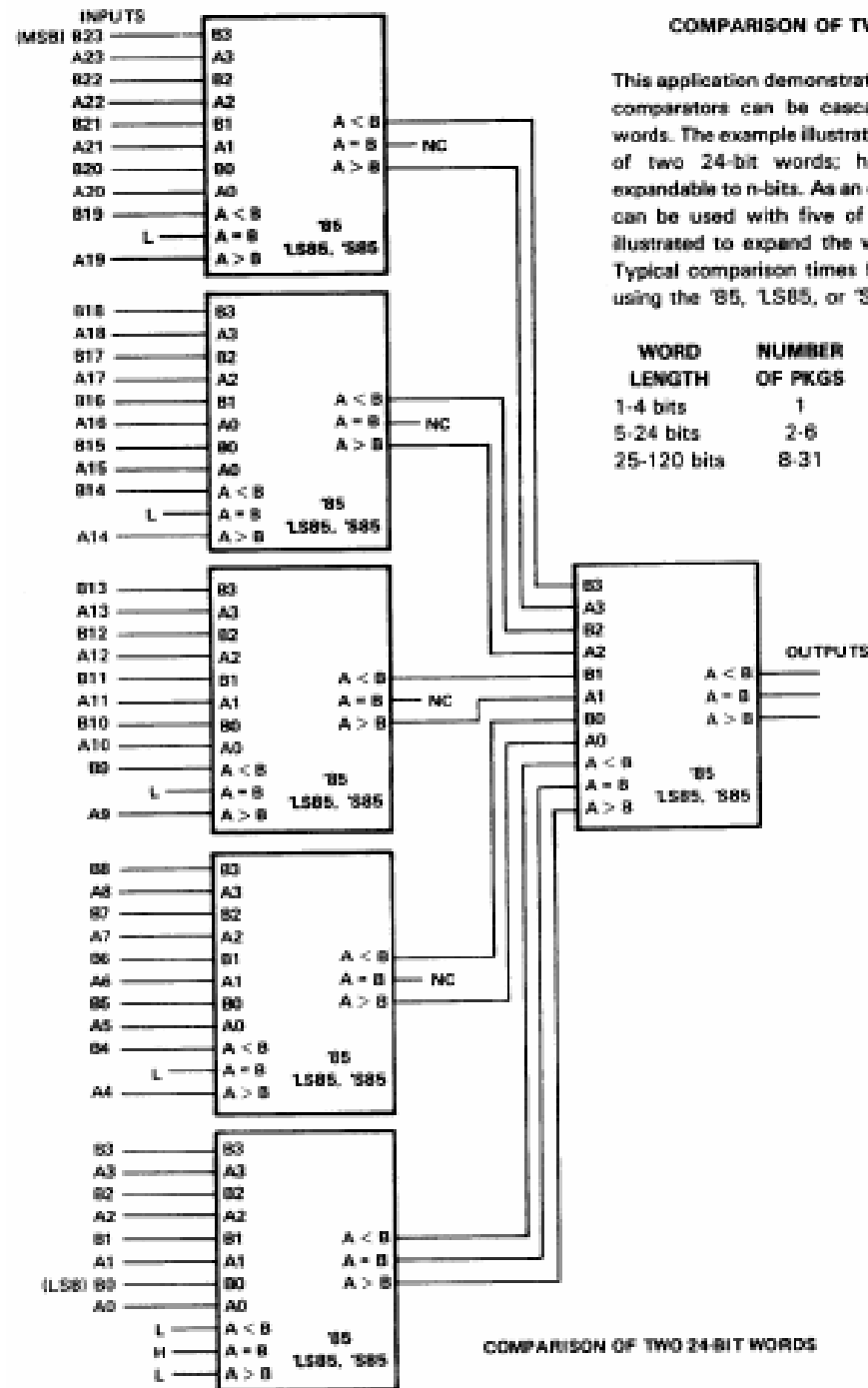


Circuito XNOR per l'eguaglianza tra A_n e B_n

Nel caso del 7485:



TYPICAL APPLICATION DATA



Operazioni con i numeri binari

Un preliminare indispensabiledi cui avevamo già parlato!

Sistemi di numerazione e codifiche:

1. Sistema decimale:

- **Base:** insieme di simboli (10)
- **Convenzione:** sistema posizionale pesato
- **Peso:** potenze della base

Esempi:

$$4532 = 4 \cdot 10^3 + 5 \cdot 10^2 + 3 \cdot 10^1 + 2 \cdot 10^0$$

$$862.39 = 8 \cdot 10^2 + 6 \cdot 10^1 + 2 \cdot 10^0 + 3 \cdot 10^{-1} + 9 \cdot 10^{-2}$$

In base "R":

$$N = C_{n-1} \cdot R^{n-1} + C_{n-2} \cdot R^{n-2} + \dots + C_{-1} \cdot R^1$$

$$N : C_{n-1} C_{n-2} \dots, C_{-1} C_{-2} \dots$$

2. Sistema ottale:

- **Base:** 8 con simboli da 0 a 7
- **Convenzione:** sistema posizionale pesato
- **Peso:** 8^n

Esempi:

$$(705)_8 = 7 \cdot 8^2 + 0 \cdot 8^1 + 5 \cdot 8^0$$

$$(501,01)_8 = 5 \cdot 8^2 + 0 \cdot 8^1 + 1 \cdot 8^0 + 0 \cdot 8^{-1} + 1 \cdot 8^{-2}$$

3. Sistema esadecimale:

- **Base:** 16 con simboli da 0 a 15 (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)
- **Convenzione:** sistema posizionale pesato
- **Peso:** 16^n

Esempi:

$$(705)_{16} = 7 \cdot 16^2 + 0 \cdot 16^1 + 5 \cdot 16^0$$

$$(501,01)_{16} = 5 \cdot 16^2 + 0 \cdot 16^1 + 1 \cdot 16^0 + 0 \cdot 16^{-1} + 1 \cdot 16^{-2}$$

2. Sistema binario:

- **Base:** 2
- **Convenzione:** sistema posizionale pesato
- **Peso:** 2^n

Esempi:

$$(110)_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

$$(101,01) = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2}$$

Regole pratiche:

- **Somma**
 $0 + 0 = 0$
 $0 + 1 = 1 + 0 = 1$
 $1 + 1 = 0$ con riporto di 1
- **Sottrazione**
 $0 - 0 = 0$
 $1 - 0 = 1$
 $1 - 1 = 0$
 $0 - 1 = 1$ con richiamo di 1

Esempi:

$$(1101)_2 + (101010)_2 = ?$$

$$(11)_2 + (1011)_2 + (111)_2 + (1011)_2 = ?$$

$$(11011)_2 - (01101)_2 = ?$$

$$(11011001)_2 - (10101010)_2 = ?$$

Ma la sottrazione è complicata !!! -> complementazione

1. Decimale:

- a 9: $129 \Rightarrow 870$
- a 10: $129 \Rightarrow 871$

2. Binaria:

- a 1: $1011 \Rightarrow 0100$
- a 2: $1011 \Rightarrow 0101$

Abbiamo un'alternativa per fare le sottrazioni:

1. DECIMALE:

i. Complemento a 10

- Eseguire il complemento a 10 del sottraendo
- Sommare il punto a. al minuendo
- Minuendo maggiore del sottraendo?

si

no

d. Scartare l'eventuale riporto

d. Complementare a 10 + segno neg.

Esempi:

$$\begin{array}{r} 547 - \\ 129 = \end{array} \rightarrow \text{complemento a 10} \rightarrow \begin{array}{r} 547 + \\ 871 = \\ \hline \end{array}$$

Lo buttiamo $\rightarrow$ 4 18

$$\begin{array}{r} 129 - \\ 547 = \end{array} \rightarrow \text{complemento a 10} \rightarrow \begin{array}{r} 129 + \\ 453 = \\ \hline \end{array}$$

complemento a 10

$$417 + 1 \rightarrow \text{cambia di segno} \rightarrow \boxed{-418}$$

2. BIANARIA:

i. Complemento a 2

- a. Eseguire il complemento a 2 del sottraendo
- b. Sommare il punto a. Al minuendo

c. Il minuendo è maggiore del sottraendo?

SI:

d. ignorare il riporto

No

- d. Complementare il risultato a 1
- e. Sommare 1 al LSB

$$\begin{array}{r} 1100 - \\ 0011 = \end{array} \rightarrow \text{compl. (2)} \rightarrow \begin{array}{r} 1100 + \\ 1101 = \\ \hline 1001 \end{array}$$

Lo buttiamo

1001

$$\begin{array}{r} 00101 - \\ 11011 = \end{array} \rightarrow \text{compl. (2)} \rightarrow \begin{array}{r} 00101 + \\ 00101 = \\ \hline 01010 \end{array}$$

complemento a 1

$10101 + 1 \rightarrow \text{chs} \rightarrow$

-10110

ii. Complemento a 1

- Eseguire il complemento a 1 del sottraendo
- Sommare il risultato di a. al minuendo

c. Il minuendo è maggiore del sottraendo?

SI:

d. Sommare il riporto a LSB (E.A.C.)

No

d. Complementare il risultato e CHS

$$\begin{array}{r}
 1110 - \\
 0110 = \\
 \hline
 \end{array}
 \rightarrow \text{compl.}(1) \rightarrow
 \begin{array}{r}
 1110 + \\
 1001 = \\
 \hline
 10111 \\
 \boxed{1} \rightarrow 1 \\
 \hline
 1000
 \end{array}$$

Lo riportiamo (EAC):

1000

$$\begin{array}{r}
 0110 - \\
 1110 = \\
 \hline
 \end{array}
 \rightarrow \text{compl.}(1) \rightarrow
 \begin{array}{r}
 0110 + \\
 0001 = \\
 \hline
 0111
 \end{array}$$

$\boxed{\text{compl.}(1) \text{ e CHS}}$

-1000

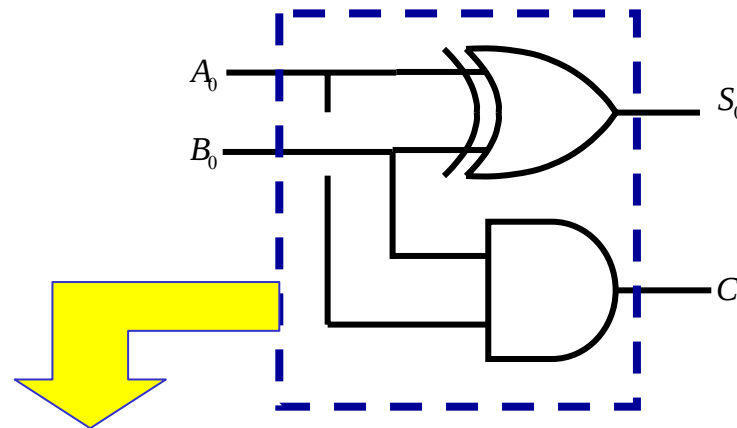
Sommatori e Sottrattori



Se dobbiamo produrre un circuito per la somma dobbiamo determinare la funzione logica tramite la relativa tavola della verità.

Nel caso di **2 bit**:

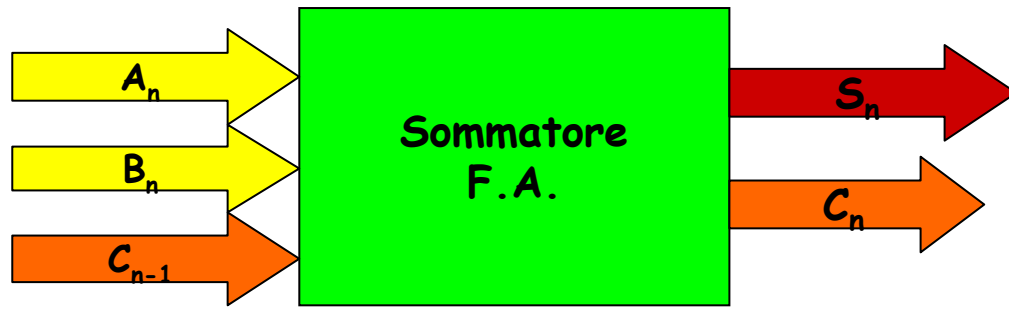
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Si chiama mezzo sommatore (Half Adder) perchè?

Non tiene conto dell' eventuale riporto in ingresso!

Sommatori e Sottrattoti



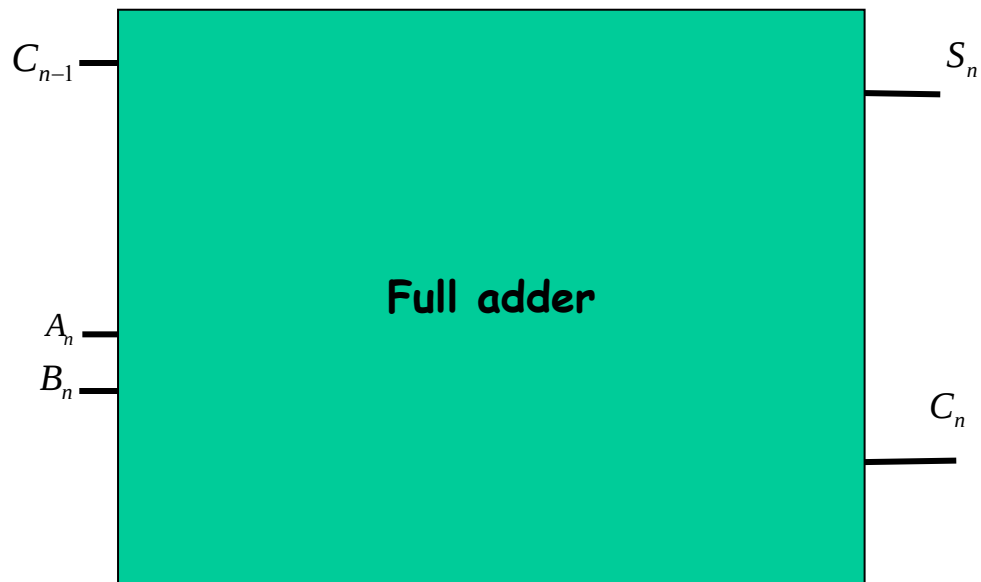
C_{n-1}	A_n	B_n	C_n	S_n	C_{n1}	C_{n2}
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	0	1	0	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	1	1	0	0	1
1	1	0	1	0	0	1
1	1	1	1	1	1	0

$$C_n = \overline{C_{n-1}} \underline{A_n B_n} + \overline{C_{n-1}} \overline{A_n} \underline{B_n} + C_{n-1} \underline{A_n} \overline{B_n} + C_{n-1} A_n B_n$$

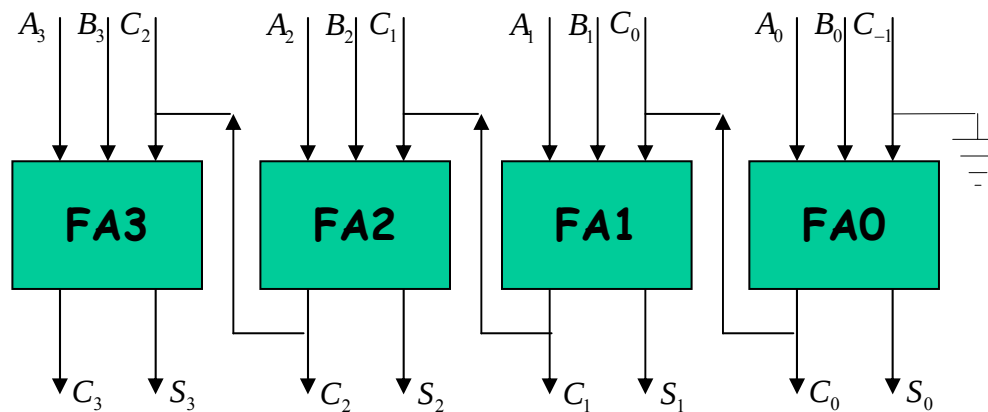
$$S_n = \overline{C_{n-1}} A_n B_n + \overline{C_{n-1}} A_n \overline{B_n} + C_{n-1} \overline{A_n} B_n + C_{n-1} A_n B_n$$

$$C_n = (\overline{C_{n-1}} + C_{n-1}) A_n B_n + C_{n-1} (\overline{A_n} B_n + A_n \overline{B_n}) = A_n B_n + C_{n-1} (A_n \oplus B_n)$$

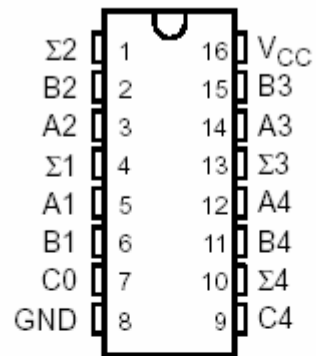
$$S_n = \overline{C_{n-1}} (\overline{A_n} B_n + A_n \overline{B_n}) + C_{n-1} (\overline{A_n} \overline{B_n} + A_n B_n) = C_{n-1} \oplus (A_n \oplus B_n)$$



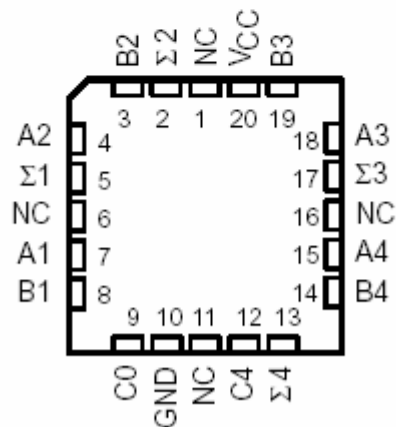
Sommatore binario parallelo a 4 bit realizzato con 4 sommatore completi in cascata:



SN54F283 . . . J PACKAGE
SN74F283 . . . D OR N PACKAGE
(TOP VIEW)

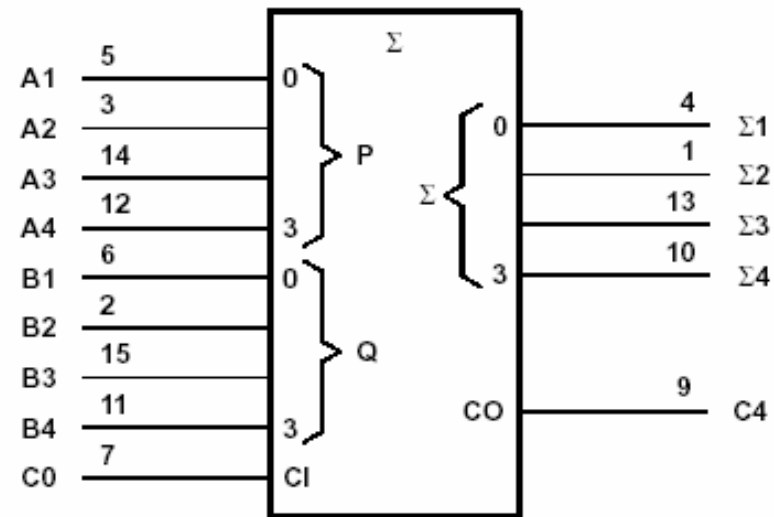


SN54F283 . . . FK PACKAGE
(TOP VIEW)



NC – No internal connection

SN54F283, SN74F283 4-BIT BINARY FULL ADDERS WITH FAST CARRY



logic diagram (positive logic)

Riporto esterno

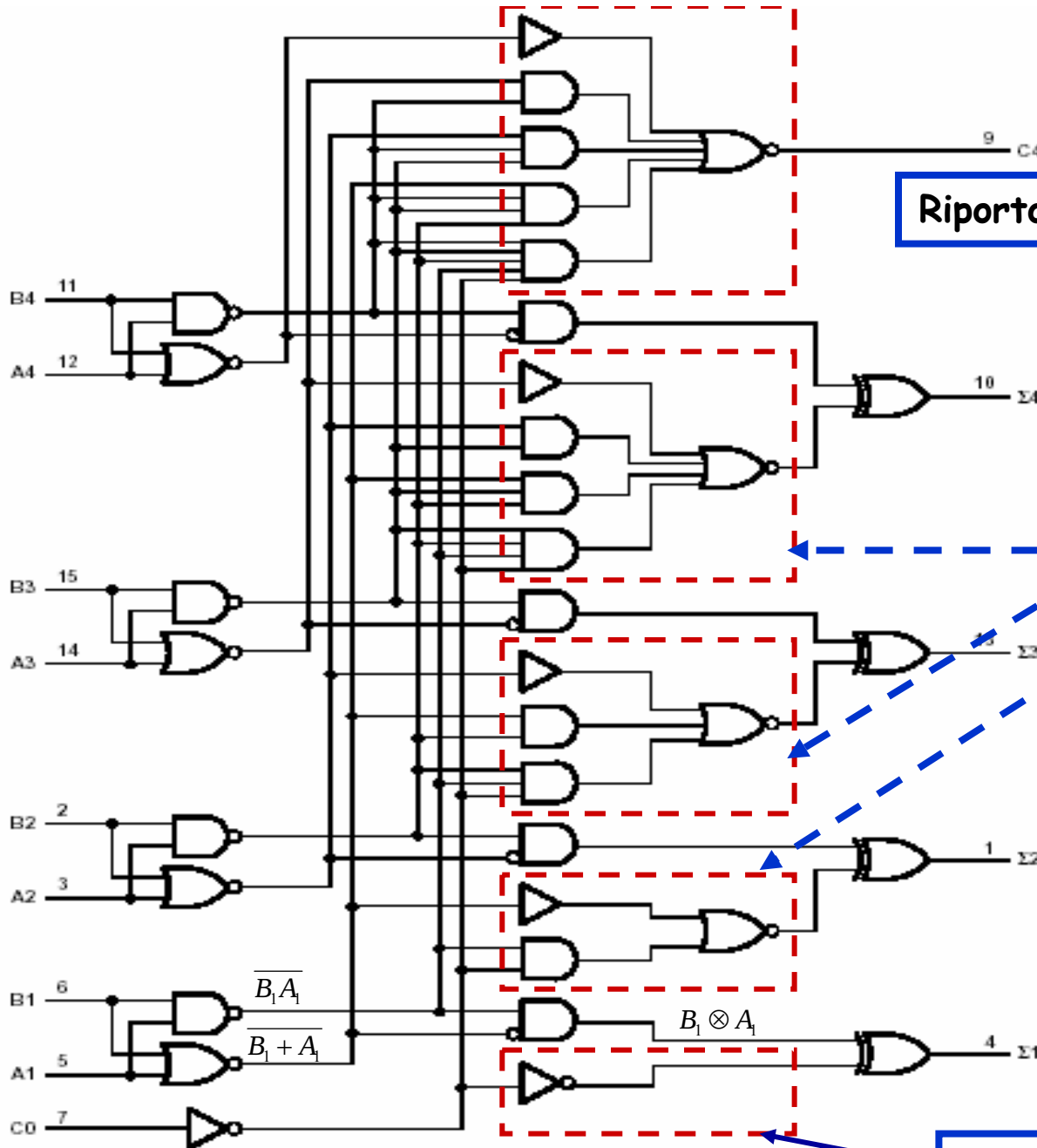
SN54F283, SN74F283
4-BIT BINARY FULL ADDERS
WITH FAST CARRY

Riporti interni

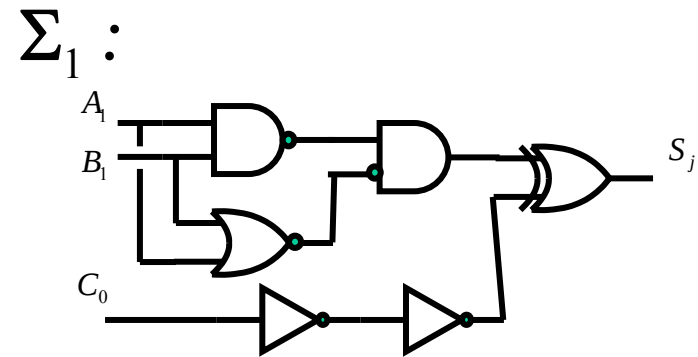
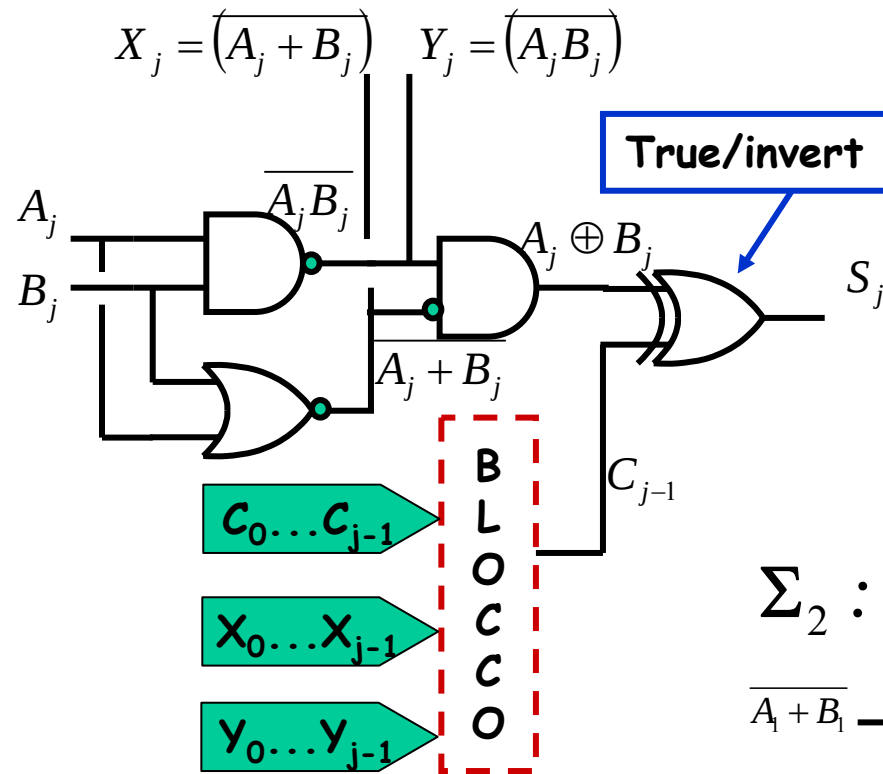
I bit S_j della somma:

- ☐ circuiti con identica parte iniziale e finale
- ☐ passando da j a $j+1$ si ha una AND in più (+ un ingresso)
- ☐ il blocco fornisce il riporto

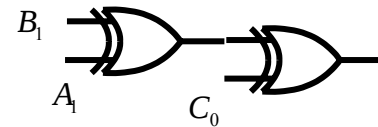
Riporto precedente



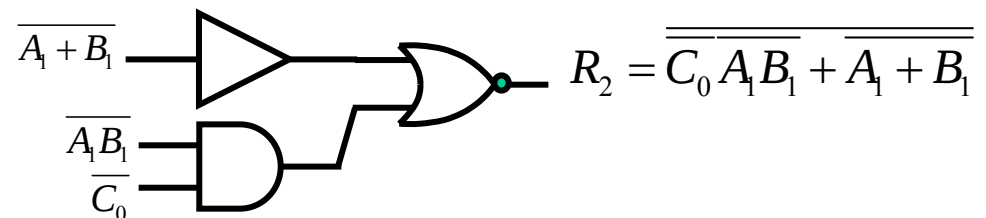
Il FULL ADDER con fast carry è formato da più moduli di questo tipo:



È equivalente a:



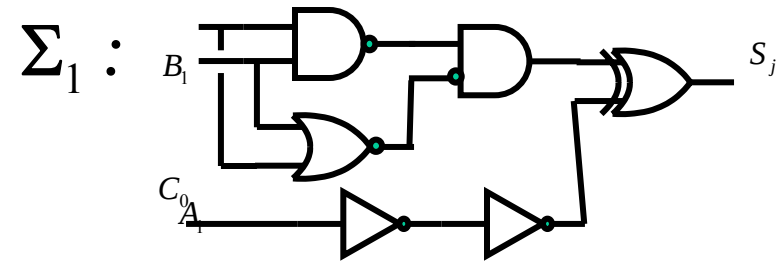
$\Sigma_2 :$ L' unica differenza è il blocco seguente:



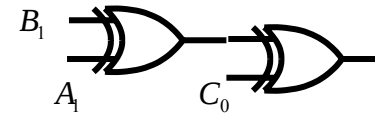
Con R_2 che si aggiunge alla somma di $A_2 B_2$ ma:

$$\begin{aligned}
 R_2 &= \overline{\overline{C_0 A_1 B_1}} (A_1 + B_1) = (C_0 + A_1 B_1) \cdot (A_1 + B_1) \\
 &= C_0 A_1 + C_0 B_1 + A_1 B_1 = C_0 (A_1 + B_1) + A_1 B_1
 \end{aligned}$$

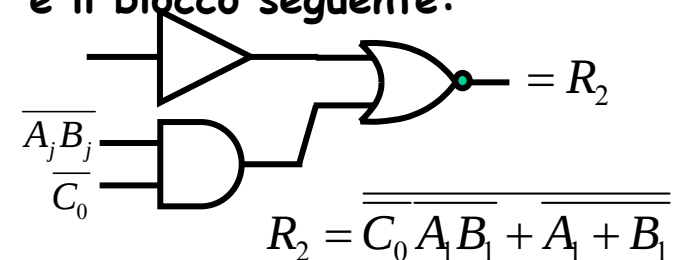
C_0	A_1	B_1	R	$A_1 \cdot B_1$	$A_1 + B_1$	$C_0(A_1 + B_1)$
0	0	0	0	0	0	0
0	0	1	0	0	1	0
0	1	0	0	0	1	0
0	1	1	1	1	0	0
1	1	0	1	0	1	1
1	0	1	1	0	1	1
1	0	0	0	0	0	0
1	1	1	1	1	1	1



È equivalente a:



Σ_2 : L' unica differenza
è il blocco seguente:



Con R_2 che si aggiunge alla somma di A_2 e B_2 ma:

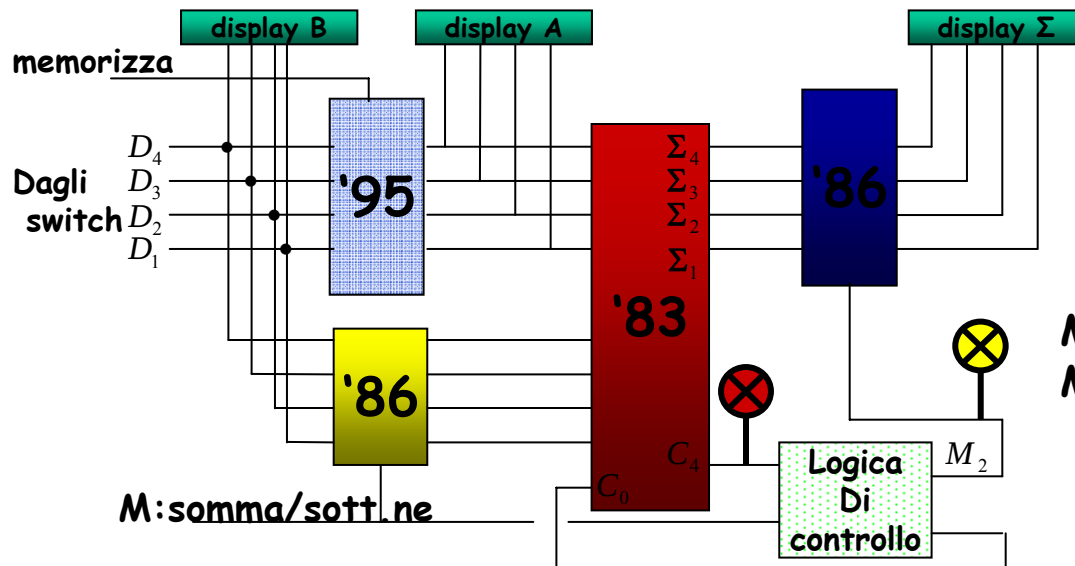
$$\begin{aligned}
 R_2 &= \overline{C_0 A_1 B_1} (A_1 + B_1) = (C_0 + A_1 B_1) \cdot (A_1 + B_1) \\
 &= C_0 A_1 + C_0 B_1 + A_1 B_1 = C_0 (A_1 + B_1) + A_1 B_1
 \end{aligned}$$

Che è vera sse sono veri ol' uno ($A_1=B_1=1$) o
L' altro: $A_1 \circ B_1 = 1$ e c' è un riporto precedente ($C_0=1$)

Esperienza D-5

Comprende le prove:

- Uso di un sommatore a 4 bit in unione ad una memoria per:
 - Addizionare numeri binari
 - Sottrarre numeri binari
 - Realizzare un collegamento E.A.C.



M	C ₄	C ₀	M ₂
0	X	0	0
1	0	0	1
1	1	1	0

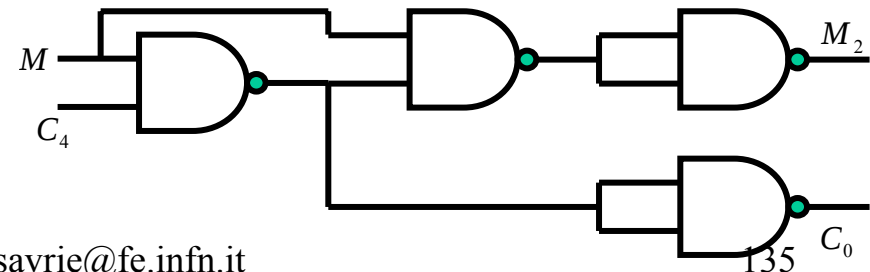
M=0 -> somma -> C₄ è un vero riporto

M=1 -> sottrazione:

- se C₄=1 -> C₀=1 no complemento
- se C₄=0 -> C₀=0 e si complementa

$$C_0 = MC_4 = \overline{MC_4}$$

$$M_2 = M(\overline{M} + \overline{C_4}) = M \cdot \overline{MC_4}$$



Esperienza D-6

Comprende le prove:

- Uso di un MUX 16 su 1 : 74150
- Uso di un contatore sincrono :74161 (contatore e memoria)
- Realizzazione di trasmissione dati da tastiera

Tastiera: tastierino numerico semplice a contatti normalmente aperti con numeri da 0 a 9

Il circuito deve:

- Segnalare se è stato premuto un tasto
- A tasto premuto generare il codice binario corrispondente

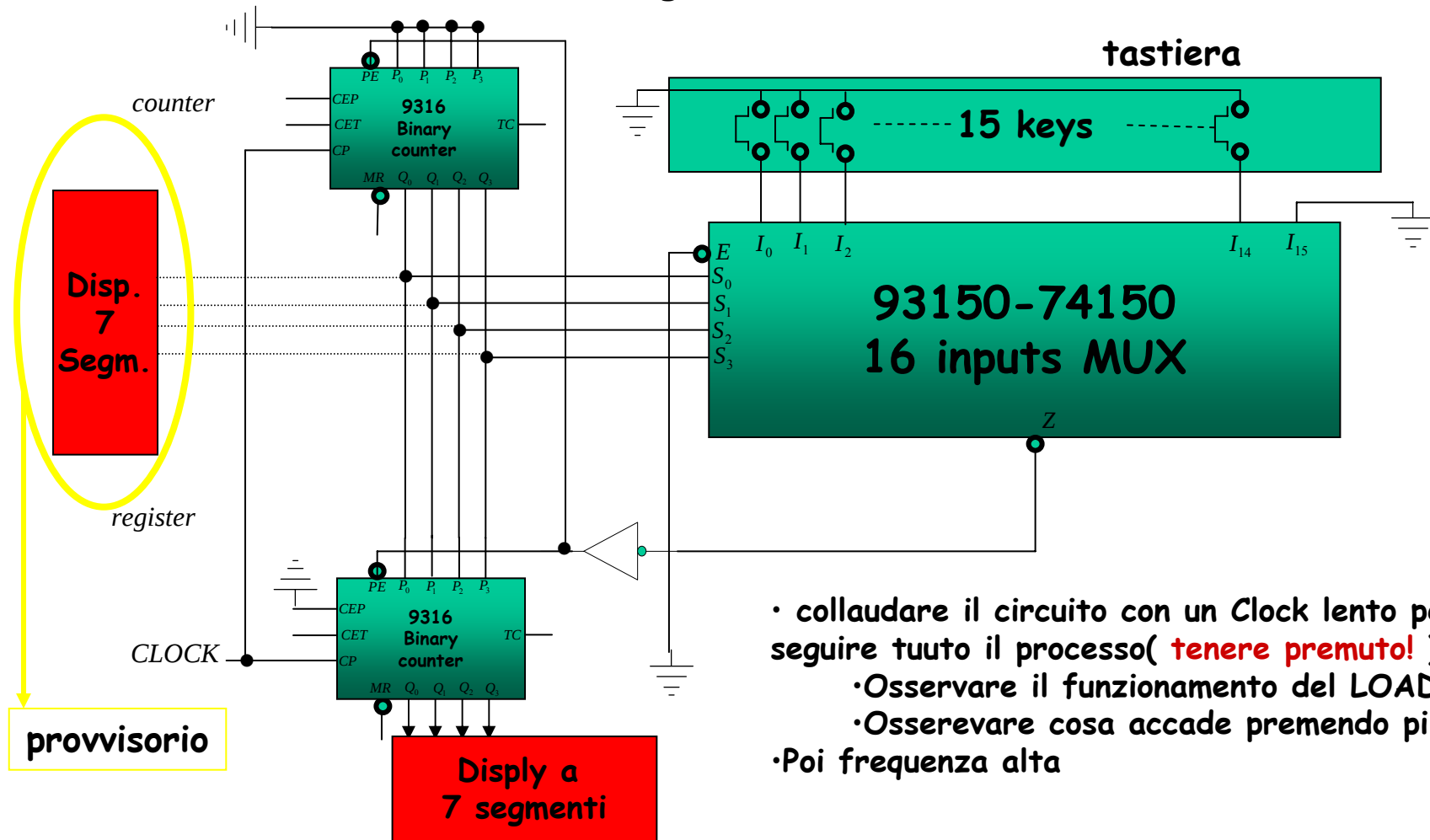
Il circuito più semplice è rappresentato da un "codificatore" (circuito combinatorio)



La nostra soluzione è più complessa ma
Più elegante e più didattica

Useremo:

1. Un MUX a 16 inputs e 4 bit di selezione
2. Tastierino numerico
3. Contatore sincrono presettabile
4. Tecnica dello "scanning"



- collaudare il circuito con un Clock lento per seguire tutto il processo(**tenere premuto!**)
- Osservare il funzionamento del LOAD
- Osservare cosa accade premendo più tasti
- Poi frequenza alta

Breve cenno sulle uscite delle porte logiche

Ruolo dell' alimentazione

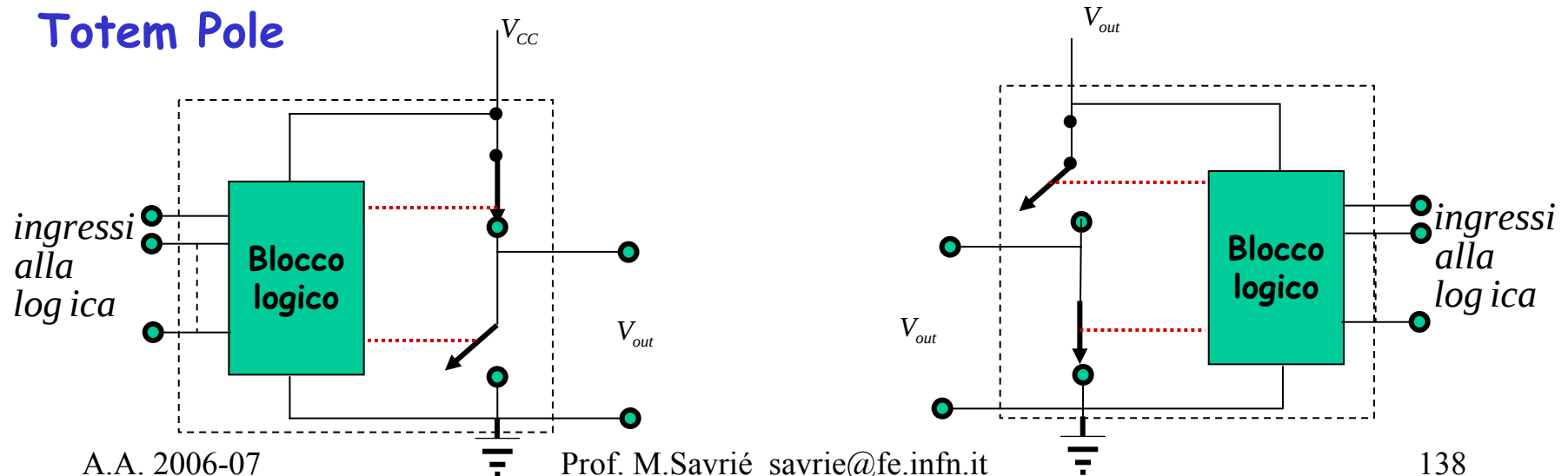
- Fornisce i livelli "alti" di uscita
- Fornisce la potenza per pilotare i circuiti di "carico" (fan-out)
- Non costituisce un segnale

I livelli sono ottenuti mediante dispositivi (transistor) che funzionano come "interruttori" comandati. In questi gli stati:

- di conduzione (interruttore chiuso)
- di non conduzione (interruttore aperto)

Sono determinati Dallo stato di una variabile di comando elettrica.

Totem Pole



Il "blocco logico":

- decide, in base alla logica (AND, OR,) se l' uscita deve essere alta o bassa

Le linee tratteggiate:

- rappresentano una "immaginaria connessione meccanica" che predispone lo stato degli interruttori

Totem pole

- gli interruttori sono sempre in stati opposti
- uscita collegata a V_{cc} -> stato **alto**
- uscita collegata a Gnd -> stato basso

I Transistor sono "interruttori imperfetti":

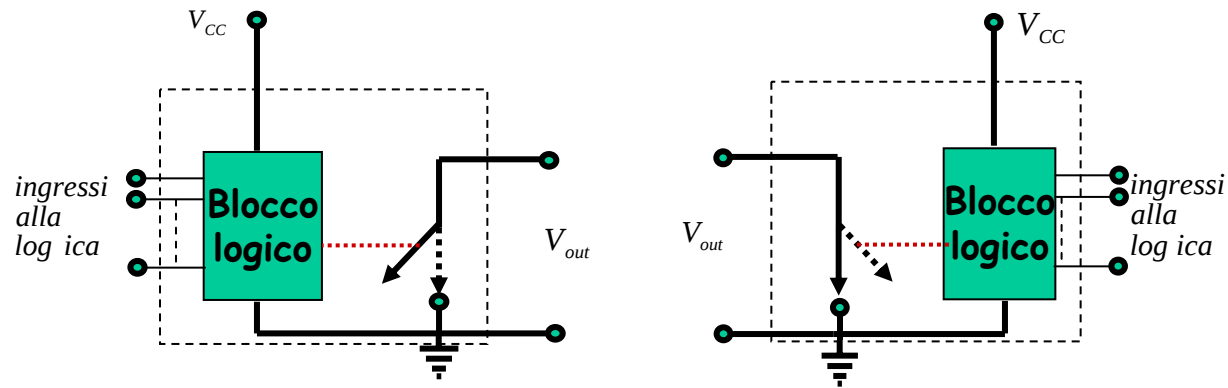
- lo stato basso non è zero
- lo stato alto non è 5V

lo stato con entrambi gli interruttori chiusi non è ammissibile!

L' integrato è predisposto per evitare quella configurazione ma bisogna evitare di:

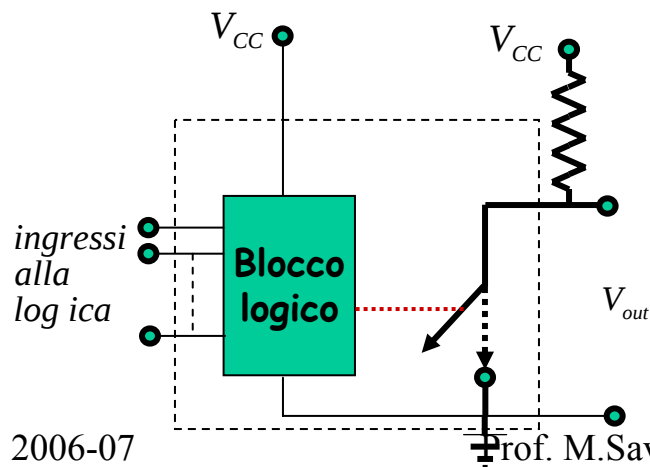
- collegare più uscite insieme
- collegare uscite con switch logici

Open Collector



I due stati dell' uscita sono **"A conduzione verso massa"**

- Tensione di uscita "prossima a zero"
- Tensione di uscita come un "filo sconnesso"
- Resistore di **pull up** e **wired AND** (AND cablato)

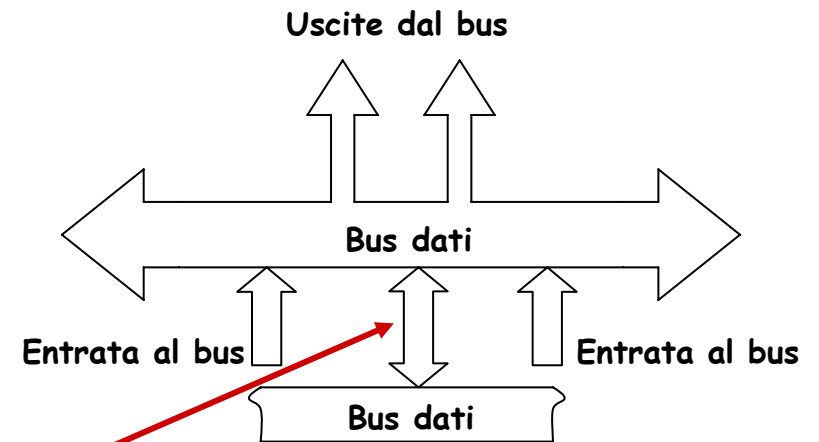


In questo caso l' alimentazione ha un ruolo anche nella logica!
Vedere gli integrati '01 e '03

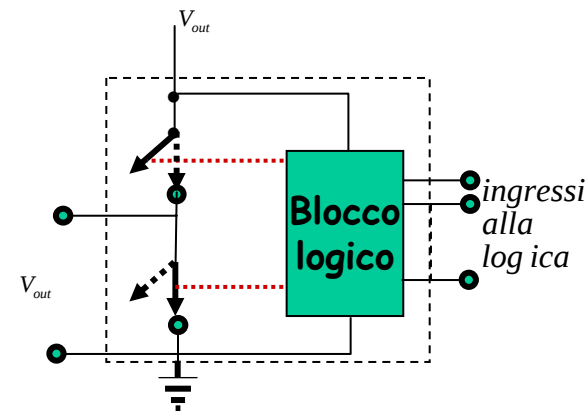
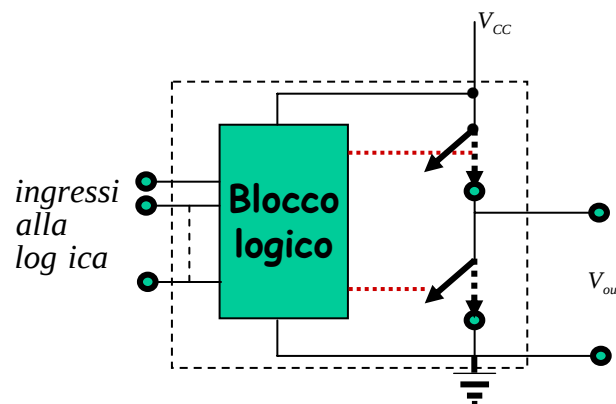
Three State

Nei sistemi complessi:

- trasferimento di dati a molti bit
- trasferimento di dati tra più blocchi
- improponibile un sistema 1bit=1filo!!
- necessità di condividere le linee
- interfacciamento bus di dati/ utenze



Collegamento bi-direz. al bus



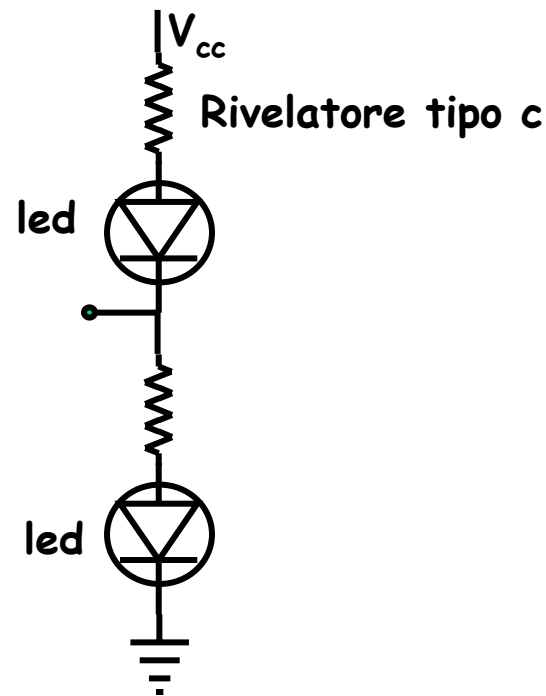
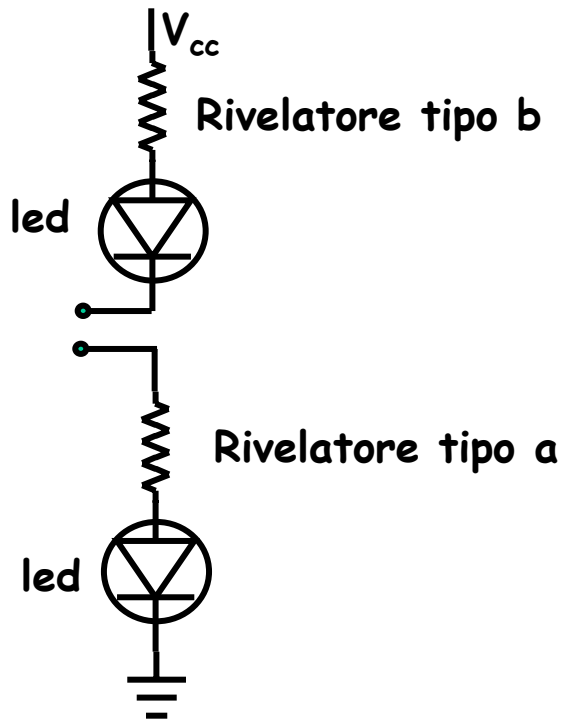
Una uscita 3-state può assumere gli stati:

1. Della Totem pole
2. Uscita "Z" detta anche ad "alta impedenza"

Esperienza D-7

Comprende le prove:

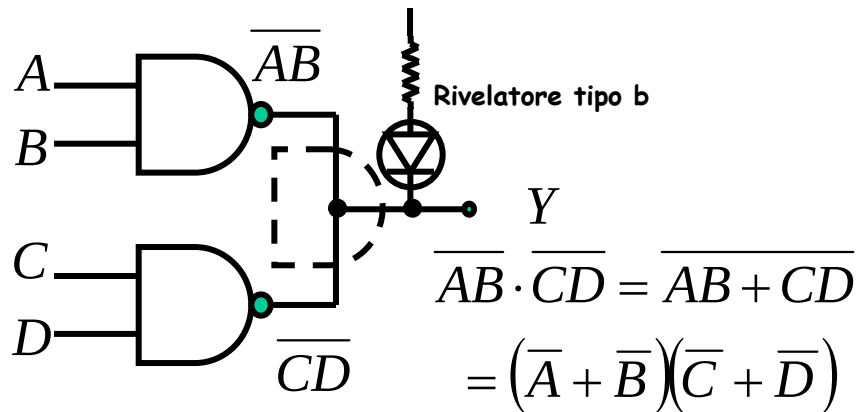
- Uso di una 7401: quadrupla NAND "open collector"
- Realizzazione di AND cablato
- Uso di dispositivi LED attivi bassi e attivi alti e loro combinazione per esaminare lo stato delle uscite degli O.C.
- Uso dell' integrato 74240
- Comunicazione tramite bus 3-state



Come si procede:

1. Rilevare il funzionamento di una porta 7401 usando
 - Un DMM misurare la tensione in uscita in corrispondenza dei livelli e senza "rivelatore"
 - Collegare il "rivelatore e ripetere la prova
 - Ripetere le prove con il rivelatore a e poi c

2. Realizzare una connessione wired AND con la '01 e verificarne il funzionamento con un rivelatore di tipo b (pull-up)

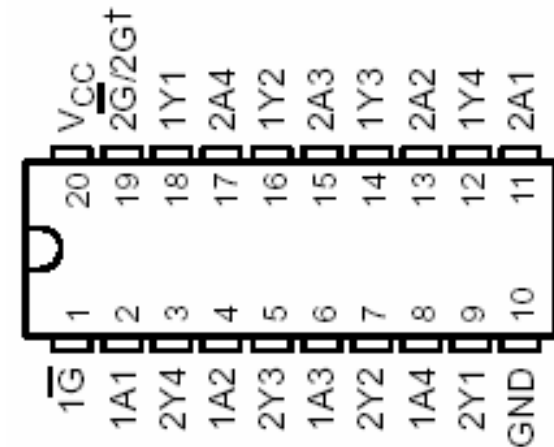


A	B	C	D	Y
0	x	0	x	1
0	x	x	0	1
x	0	x	0	1
x	0	0	x	1
1	1	x	x	0
x	x	1	1	0

3. Collaudare un buffer 74240 usando il DMM e tutti i rivelatori visti:

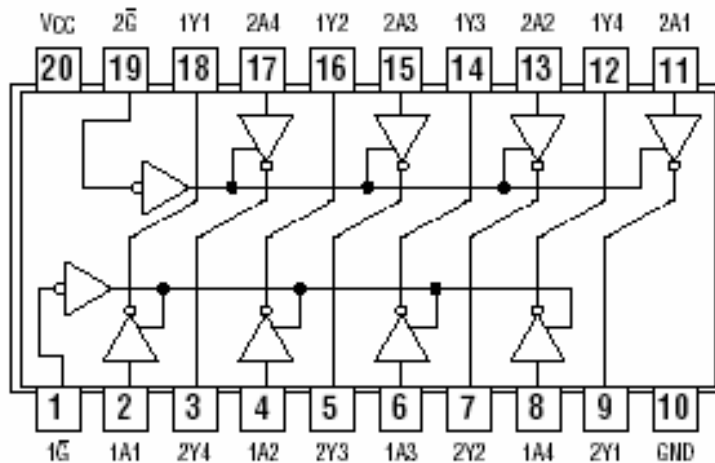
- Pilotare con gli switch i dati in ingresso e l'abilitazione G

SN54LS', SN54S' ... J OR W PACKAGE
SN74LS240, SN74LS244 ... DB, DW, N, OR NS PACKAGE
SN74LS241 ... DW, N, OR NS PACKAGE
SN74S' ... DW OR N PACKAGE
(TOP VIEW)

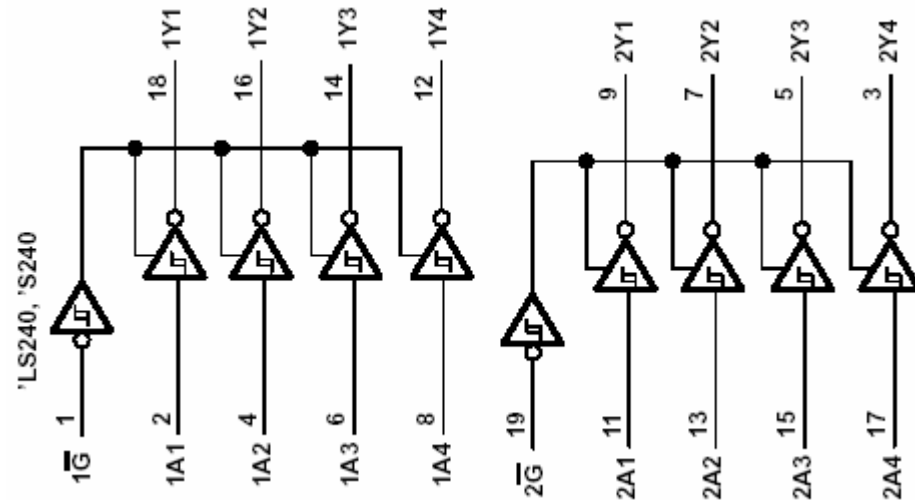


240

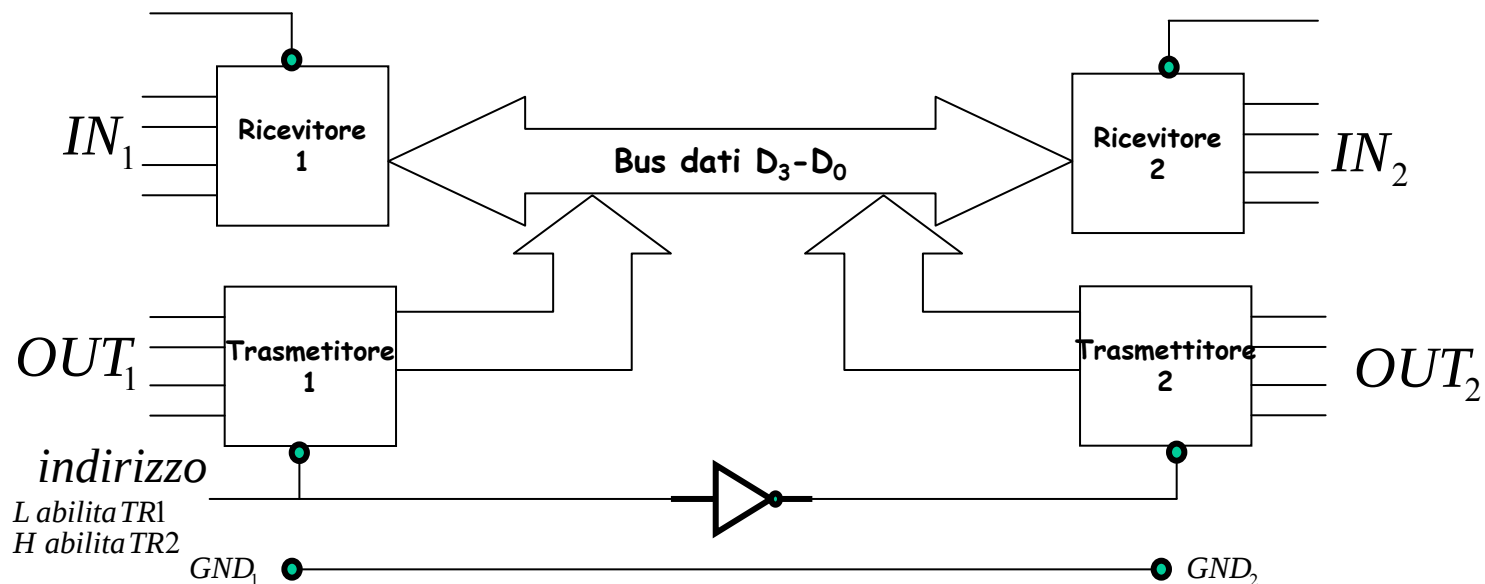
OCTAL BUFFERS/LINE DRIVERS/LINE RECEIVERS



logic diagram



4. Uso del buffer 3-state per la trasmissione bidirezionale di dati a 4 bit tramite bus di comunicazione e linea di indirizzo che abilita una stazione alla volta.



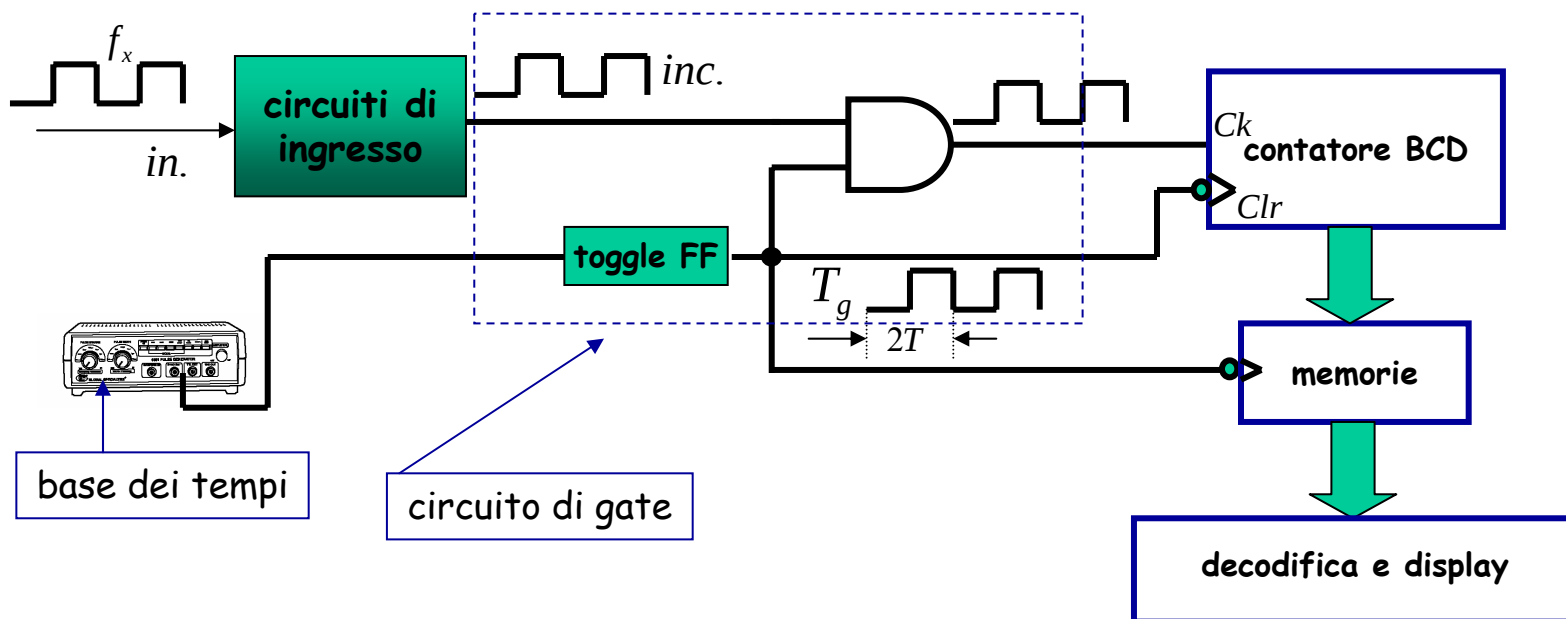
Due stazioni ricetrasmittenti (due gruppi frontali) comunicano attraverso un BUS di dati. Ciascuna è identificata da un codice di indirizzo (0=master;1=Slave) ed è abilitata a trasmettere solo quando il bit di indirizzocorrisponde al suo codice. La linea GND_1-GND_2 **è fondamentale!!!**

Esperienza D-8

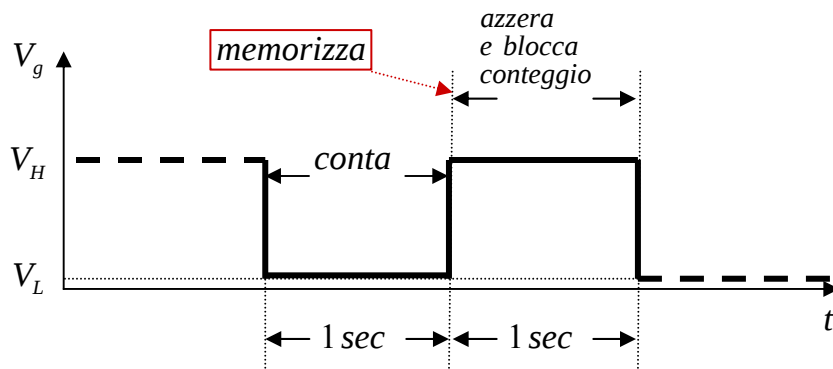
Comprende le prove:

- Realizzazione di un contatore di impulsi a tre cifre
 - Suo impiego per misurare la frequenza
 - Suo impiego per misure di periodo

Il contatore viene impiegato per misurare un rapporto tra frequenze e periodi



□ Misure di frequenza:



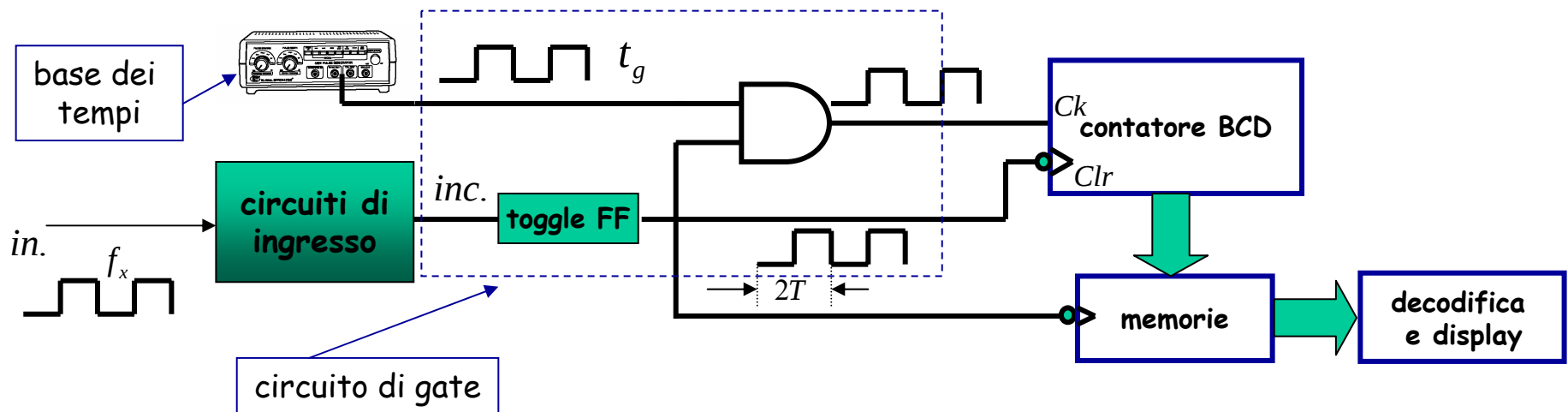
Se contiamo per un tempo t_g (=tempo di gate) pari al periodo di un segnale TTL di frequenza f_2 e gli impulsi da contare sono periodici di periodo T_1 . Il numero di impulsi contati vale:

$$f_x = \frac{N}{T_g}$$

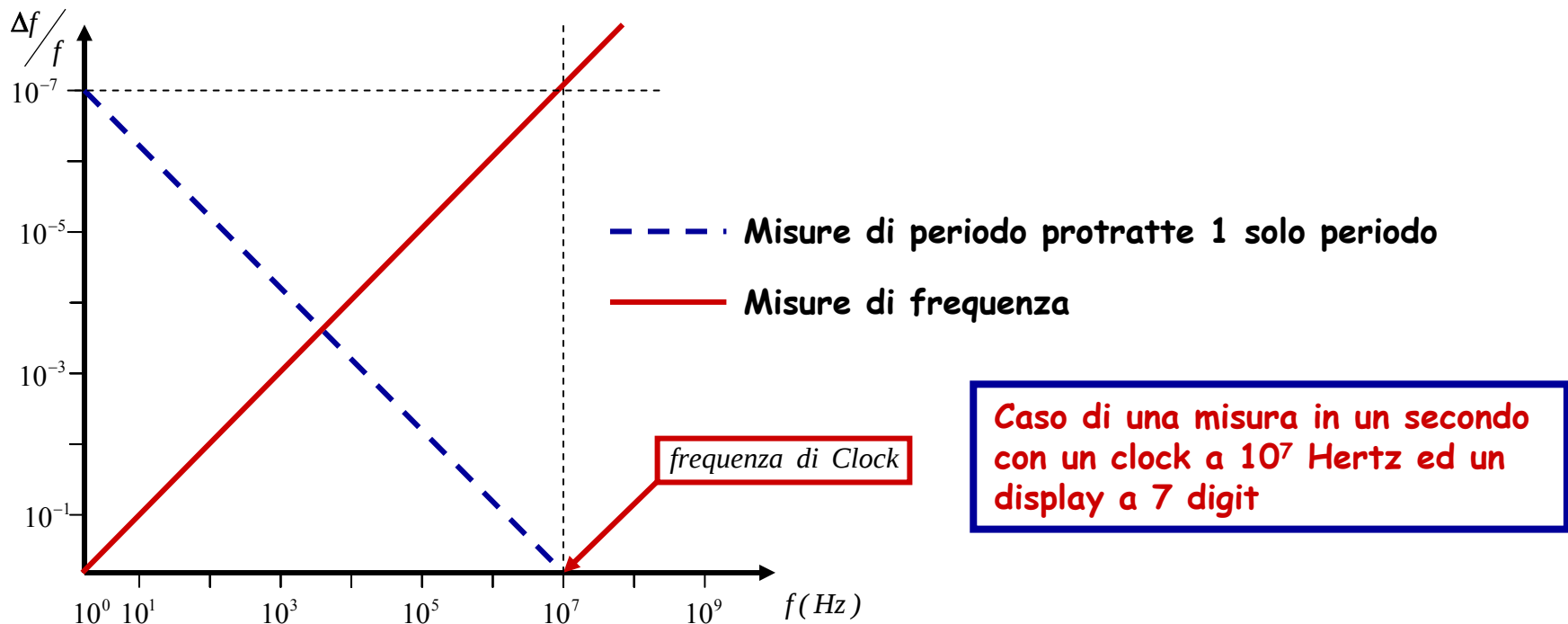
$$N = f_1 \cdot t_g = \frac{f_1}{f_2} = \frac{T_2}{T_1}$$

se $t_g = \text{tempo di gate} = 1s \rightarrow f_x = N(\text{Hz})$

□ Misure di Periodo: scambiamo i ruoli tra f_x e di gate e T_x è il periodo incognito



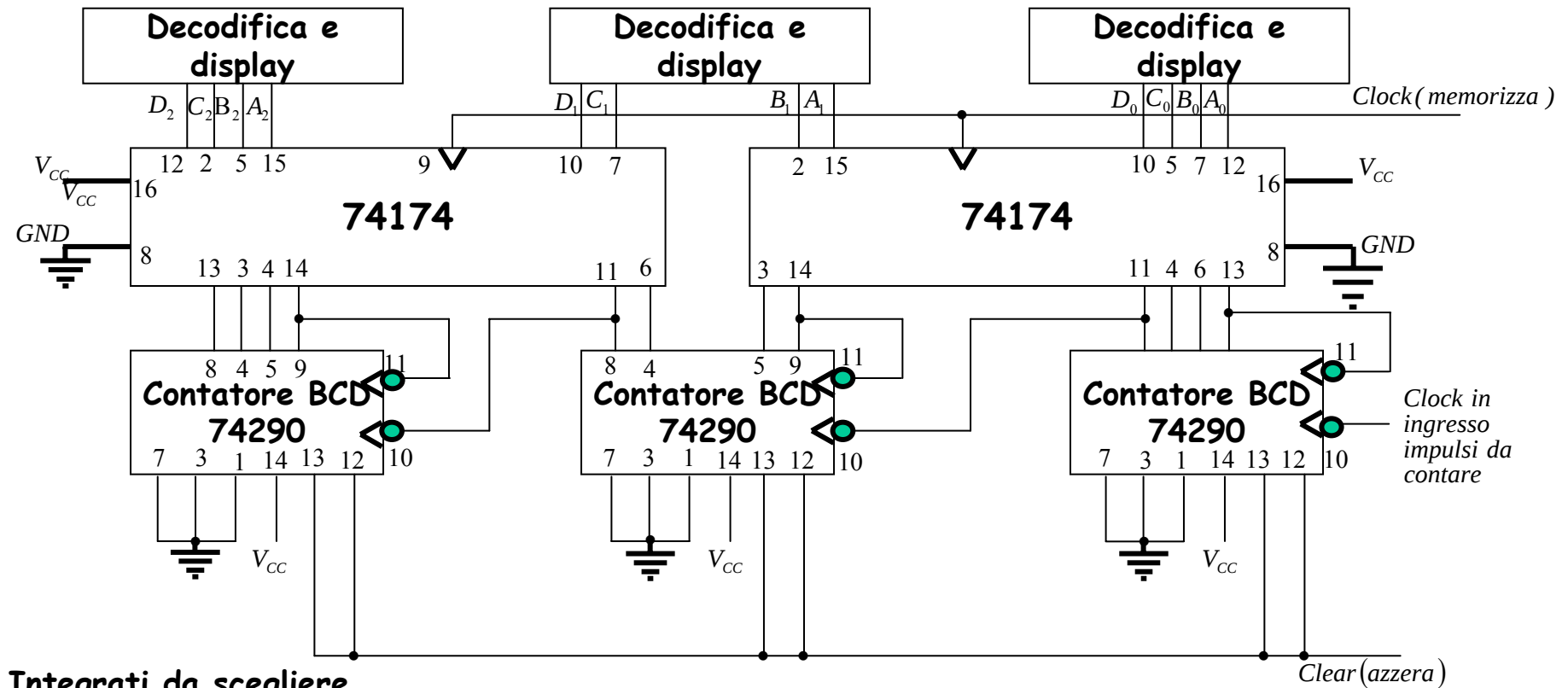
$$T_x = N t_g \Rightarrow \text{se } t_g = 1s \Rightarrow T_x = N(s)$$



- se una frequenza (periodo) è nota si risale all' altra
- indipendentemente da quale sia nota: la più alta deve essere f_1 altrimenti non si conta nemmeno un impulso
- Errori:
 - l' errore sulla frequenza nota si riflettono su quella da misurare (quarzi: $1 \cdot 10^{-9}$)
 - Errore sul tempo di gate
 - Errore di + o - un digit: è importante per definire se è conveniente misurare T of
 - Circa 0.1% a fondo scala
 - Circa 100% a inizio scala (1 conteggio)

Infatti le misure di frequenza protrate per 1 s hanno un errore di $\pm 1/f_x$ dovute a ± 1 conteggio (l' errore vale $1/N$ con N =numero di impulsi contati). frequenza bassa $\rightarrow$ conviene misurare T (N è alto). Frequenza alta $\rightarrow$ conviene misurare F(N è alto)

Contatore a tre cifre decimali con azzeramento e memorizzazione



Integrati da scegliere

- '90 e '290 differiscono solo per la piedinatura. Reset attivo alto
- '390 è un doppio contatore BCD con Clear attivo alto e 16 pin
- '174 è una memoria a 16 pin ha 6 FF tipo D edge triggered
- '273 è una memoria a 20 pin ha 8 FF tipo D edge triggered
- '75 è una 4-pla latch NON edge triggered: è trasparente quando Ck è alto
- '194 (registro) e '161 (contatore) hanno 16 pin e memorizzano 4 bit alla salita del Ck