

ALGEBRA DI BOOLE

Prof. *Antonio Quattrone*
ITIS VERONA-TRENTO - ME
a.s. 1990/91

POSTULATI FONDAMENTALI:

- 1) UNA VARIABILE BOOLEANA PUO' ASSUMERE DUE SOLI VALORI: 0, 1 (FALSO, VERO);
- 2) SI DEFINISCONO TRE OPERATORI FONDAMENTALI: NOT, AND, OR.
- 3) PER GLI OPERATORI LOGICI FONDAMENTALI VALGONO LE SEGUENTI RELAZIONI:
 - a) NOT (NEGAZIONE O COMPLEMENTAZIONE) : $\overline{1} = 0$, $\overline{0} = 1$;
O INVERSIONE

- b) AND (PRODOTTO LOGICO):

$$\begin{aligned} 0 \cdot 0 &= 0 \\ 0 \cdot 1 &= 0 \\ 1 \cdot 0 &= 0 \\ 1 \cdot 1 &= 1 \end{aligned}$$

- c) OR (SOMMA LOGICA):

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 0 &= 1 \\ 1 + 1 &= 1 \end{aligned}$$

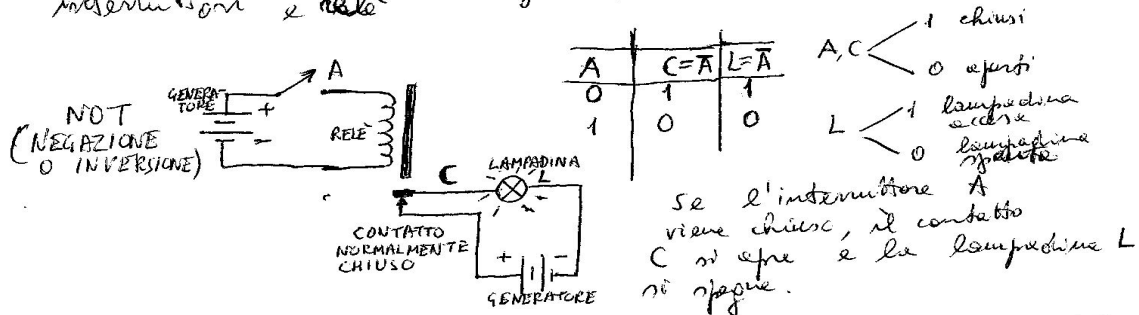
Per la verifica delle proprietà degli operatori logici fondamentali si possono utilizzare analogie elettriche o idrauliche, associando il valore 1 ad un contatto elettrico chiuso o ad una valvola aperta ed il valore 0 ad un contatto elettrico aperto o ad una valvola chiusa.

Esempi:

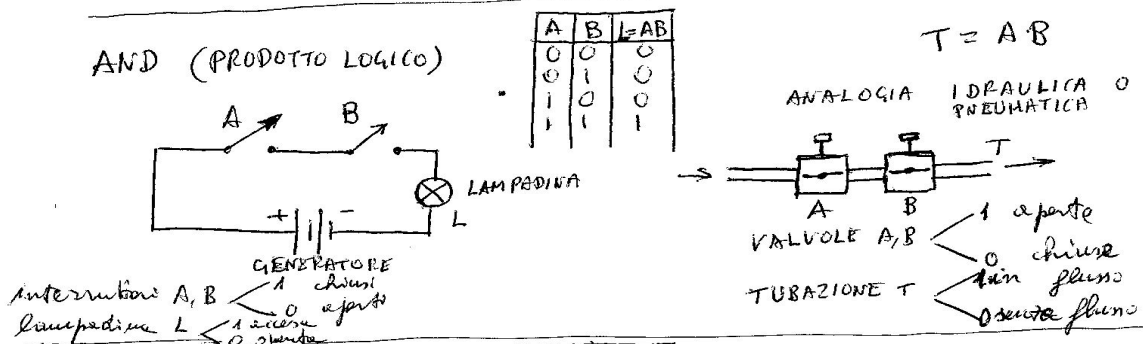
	VALORE 1	VALORE 0
frece	vera	falsa
contatto elettrico	chiuso	aperto
valvola	aperta	chiusa
rete	eccitata	diseccitata
uscita di elemento	in flusso	senza flusso
fluidico	in flusso	senza flusso
tubazione	accesa	spenta
lampadina		

Nel 1938 Shannon propose l'applicazione dell'algebra di Boole (1847) all'analisi ed al progetto dei circuiti a contatti, realizzati mediante interruttori e relè.

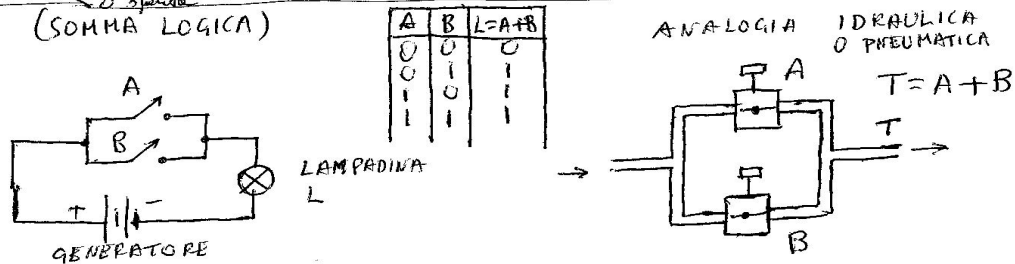
Realizzazione di circuiti logici fondamentali mediante interruttori e relè



AND (PRODOTTO LOGICO)



OR (SOMMA LOGICA)



GLI OPERATORI AND E OR SI APPLICANO A DUE O PIU' VARIABILI BOOLEANE

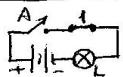
TEOREMI FONDAMENTALI

(Da ogni teorema si ricava il teorema duale sostituendo a $+$ $\cdot$ e a $\cdot$ $+$, e $0 \leftrightarrow 1$, e viceversa)

- 1) Doppia negazione o involuzione

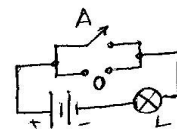
$$\overline{\overline{A}} = A \quad - \begin{cases} \overline{0} = 1 \\ \overline{1} = 0 \end{cases}$$

- 2) IDENTITÀ

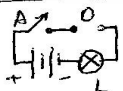


$$A \cdot 1 = A \leftrightarrow \overline{A} + 0 = A$$

$$\begin{array}{l} 1 \cdot 1 = 1 \\ 0 \cdot 1 = 0 \end{array} \leftrightarrow \begin{array}{l} 1 + 0 = 1 \\ 0 + 0 = 0 \end{array}$$

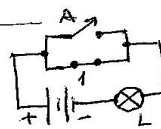


- 3)

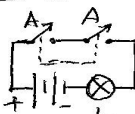


$$A \cdot 0 = 0 \leftrightarrow A + 1 = 1$$

$$\begin{array}{l} 1 \cdot 0 = 0 \\ 0 \cdot 0 = 0 \end{array} \leftrightarrow \begin{array}{l} 1 + 1 = 1 \\ 0 + 1 = 1 \end{array}$$

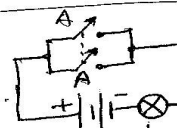


- 4) IDEMPOLENZA



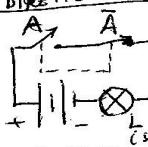
$$A \cdot A = A \leftrightarrow A + A = A$$

$$\begin{array}{l} 1 \cdot 1 = 1 \\ 0 \cdot 0 = 0 \end{array} \leftrightarrow \begin{array}{l} 1 + 1 = 1 \\ 0 + 0 = 0 \end{array}$$



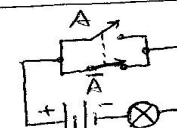
INTERUTTORI ACCOPPIATI MECCANICAMENTE IN MODO DIRETTO (si chiudono e si aprono simultaneamente)

- 5)



$$A \cdot \overline{A} = 0 \leftrightarrow A + \overline{A} = 1$$

$$\begin{array}{l} 1 \cdot 0 = 0 \\ 0 \cdot 1 = 0 \end{array} \leftrightarrow \begin{array}{l} 1 + 0 = 1 \\ 0 + 1 = 1 \end{array}$$



INTERUTTORI ACCOPPIATI MECCANICAMENTE IN MODO INVERSO (se si chiude A si apre A-bar e viceversa)

$$A \cdot B = B \cdot A \leftrightarrow A + B = B + A$$

- 6) Proprietà commutativa degli operatori AND e OR

- 7) Proprietà associativa degli operatori AND e OR

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

- 8) Proprietà distributiva degli operatori AND e OR

$$A \cdot (B + C) = AB + AC \leftrightarrow A + B \cdot C = (A + B)(A + C)$$

- 9) Teoremi di assorbimento

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

infatti: $A(1+B) = A \cdot 1 = A$ infatti: $A \cdot A + AB = A + AB = A$

- 10) Teoremi di De Morgan (validi per 2 e per n variabili)

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

per n variabili: $\overline{A+B+\dots} = \overline{A} \cdot \overline{B} \cdot \dots$ $\overline{A \cdot B \cdot \dots} = \overline{A} + \overline{B} + \dots$

OPERATORI LOGICI

4

SIMBOLI E TABELLE DI VERITA'

1) NOT $A \rightarrow Y = \bar{A}$

A	$Y = \bar{A}$
0	1
1	0

2) AND $A, B \rightarrow Y = A \cdot B$

A	B	$Y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

L'operatore AND si applica a 2 o più variabili.

$A, B, C, \dots, Z \rightarrow Y = A \cdot B \cdot C \cdot \dots \cdot Z$

L'uscita Y assume il valore 1 soltanto se tutte le variabili d'ingresso assumono il valore 1.

3) OR $A, B \rightarrow Y = A + B$

A	B	$Y = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

L'operatore OR si applica a 2 o più variabili.

$A, B, C, \dots, Z \rightarrow Y = A + B + C + \dots + Z$

L'uscita Y assume il valore 1 se una o più variabili d'ingresso assumono il valore 1.

4) NAND (PRODOTTO LOGICO NEGATO)

$A, B \rightarrow Y = \overline{A \cdot B}$

(AND + NOT)

$A, B \rightarrow Y = \overline{A \cdot B}$

$A, B, C, \dots, Z \rightarrow Y = \overline{A \cdot B \cdot C \cdot \dots \cdot Z}$

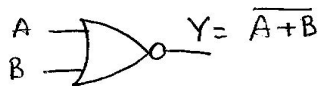
A	B	$Y = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

$A \rightarrow Y = \bar{A}$

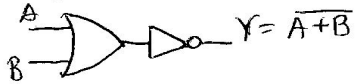
PORTA NAND UTILIZZATA COME PORTA NOT

Se l'operatore NAND si applica a più variabili, l'uscita Y assume il valore 0 soltanto se tutte le variabili d'ingresso assumono il valore 1.

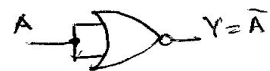
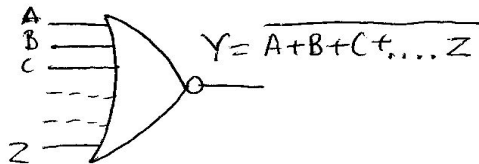
5) NOR (SOMMA LOGICA NEGATA)



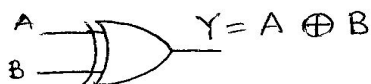
(OR + NOT)



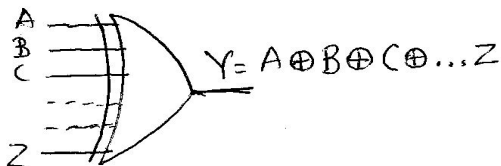
A	B	$Y = A + B$
0	0	1
0	1	0
1	0	0
1	1	0

PORTA NOR UTILIZZATA
COME PORTA NOT

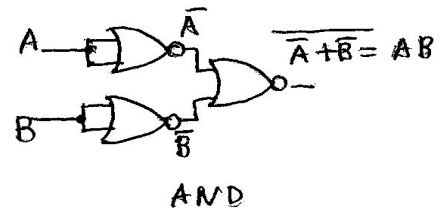
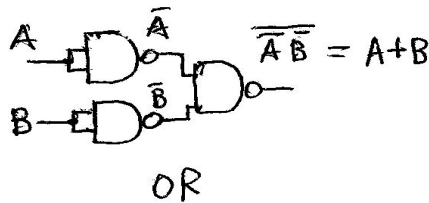
Se l'operatore NOR si applica a più variabili, l'uscita Y assume il valore 1 soltanto se tutte le variabili d'ingresso assumono il valore 0.

6) EXOR (OR ESCLUSIVO)
(SOMMA LOGICA ESCLUSIVA)

A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

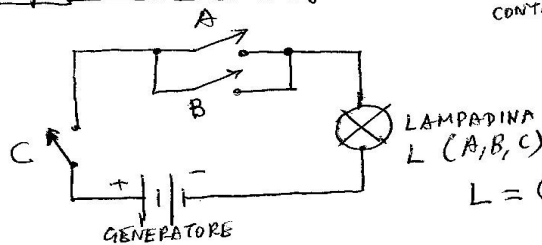


Se l'operatore EXOR si applica a più variabili, l'uscita Y assume il valore 1 soltanto se un numero dispari di variabili assume il valore 1.



Reti logiche realizzate mediante contatti e dispositivi elettronici

Esempio di deduzione di funzione booleana da un circuito elettronico



CONTATTI A, B, C $\begin{cases} \text{chiusi} \rightarrow 1 \\ \text{aperti} \rightarrow 0 \end{cases}$

LAMPADINA L $\begin{cases} \text{aperta} \rightarrow 0 \\ \text{accesa} \rightarrow 1 \end{cases}$

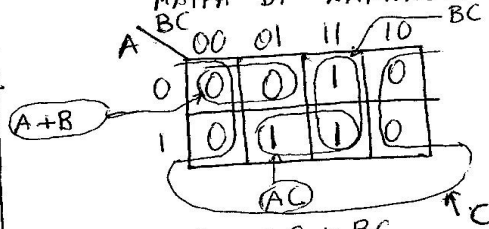
$$L = (A+B) \cdot C = A \cdot C + B \cdot C$$



TABELLA DI VERITÀ DELLA FUNZIONE BOOLEANA $L(A, B, C)$

A	B	C	L	MINTERMINI	MAXTERMINI
0	0	0	0	$P_0 = \bar{A}\bar{B}\bar{C}$	$S_0 = A+B+C$
0	0	1	0	$P_1 = \bar{A}\bar{B}C$	$S_1 = A+B+\bar{C}$
0	1	0	0	$P_2 = \bar{A}B\bar{C}$	$S_2 = A+\bar{B}+C$
0	1	1	1	$P_3 = \bar{A}BC$	$S_3 = A+\bar{B}+\bar{C}$
1	0	0	0	$P_4 = A\bar{B}\bar{C}$	$S_4 = \bar{A}+B+C$
1	0	1	1	$P_5 = A\bar{B}C$	$S_5 = \bar{A}+B+\bar{C}$
1	1	0	0	$P_6 = AB\bar{C}$	$S_6 = \bar{A}+\bar{B}+C$
1	1	1	1	$P_7 = ABC$	$S_7 = \bar{A}+\bar{B}+\bar{C}$

MAPPA DI KARNAUGH DI $L(A, B, C)$



$$L = AC + BC$$

oppure

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

FORME CANONICHE : $\begin{cases} \text{DISGIUNTIVA: } Y = \bar{A} \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot C \\ \text{CONGIUNTIVA: } Y = (A+B+C)(A+B+\bar{C})(A+\bar{B}+C)(A+\bar{B}+\bar{C}) \end{cases}$

LA FORMA CANONICA DISGIUNTIVA DI UNA FUNZIONE BOOLEANA SI OTTIENE CONSIDERANDO LA SOMMA LOGICA DEI MINTERMINI CORRISPONDENTI ALLE RIGHE IN CUI LA FUNZIONE VALE 1;

LA FORMA CANONICA CONGIUNTIVA SI OTTIENE CONSIDERANDO IL PRODOTTO LOGICO DEI MAXTERMINI CORRISPONDENTI ALLE RIGHE IN CUI LA FUNZIONE VALE 0.

MINTERMINE È OGNI PRODOTTO LOGICO DI TUTTE LE VARIABILI, CONSIDERATE IN FORMA DIRETTA (A, B, C) O NEGATA ($\bar{A}$, $\bar{B}$, $\bar{C}$); MAXTERMINE È OGNI SOMMA LOGICA DI TUTTE LE VARIABILI, CONSIDERATE IN FORMA DIRETTA (A, B, C) O NEGATA ($\bar{A}$, $\bar{B}$, $\bar{C}$).

I MINTERMINI SI OTTENGONO DALLA TABELLA DI VERITÀ ASSOCIANDO AGLI ZERI LE VARIABILI NEGATE ED AGLI UNO LE VARIABILI DIRETTE, I MAXTERMINI SI OTTENGONO ASSOCIANDO AGLI ZERI LE VARIABILI DIRETTE ED AGLI UNO LE VARIABILI NEGATE.

- 1) La fase di lavorazione viene avviata se il pezzo è posizionato correttamente (segnale S_1 al livello alto), se la pressione dell'olio nel cilindro supera il valore minimo (uscita del sensore P al livello alto) e se è in funzione il sistema di sicurezza (segnale S_2 al livello alto).

$$Y = S_1 P S_2$$

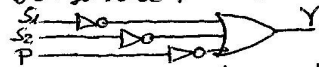
→ 1 fase di lavorazione avviata



- 2) La slitta di una macchina utensile ritorna alla posizione di partenza se il sistema di controllo cessa di funzionare (segnale S_1 al livello basso), oppure se il pezzo non è posizionato correttamente (segnale S_2 basso) ^{oppure} se la pressione dell'olio nel cilindro è inferiore al valore normale (uscita del sensore P al livello basso).

$$Y = \bar{S}_1 + \bar{S}_2 + \bar{P}$$

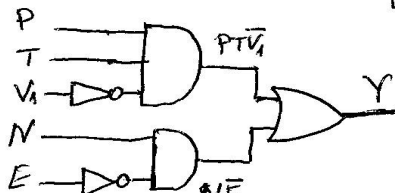
→ ritorno della slitta alla posizione di partenza



- 3) Il circuito ausiliario di raffreddamento di un reattore nucleare viene attivato se la pressione e la temperatura del fluido di raffreddamento nel circuito primario superano i valori normali (uscite dei sensori P e T al livello alto) e se l'elettrovalvola di sicurezza V_1 è chiusa, oppure se il flusso neutronico supera il valore normale (segnale N al livello alto) e l'elettropompa E del circuito primario non è in funzione (segnale E al livello basso).

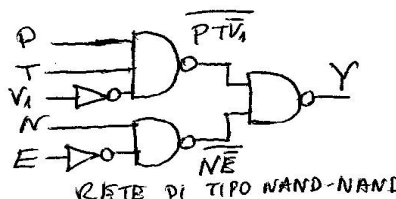
$$Y = P T \bar{V}_1 + N \bar{E}$$

→ 1 circuito ausiliario di raffreddamento in funzione



RETE DI TIPO AND-OR

$$Y = P T \bar{V}_1 + N \bar{E} = \overline{P T \bar{V}_1 \cdot N \bar{E}}$$



RETE DI TIPO NAND-NAND

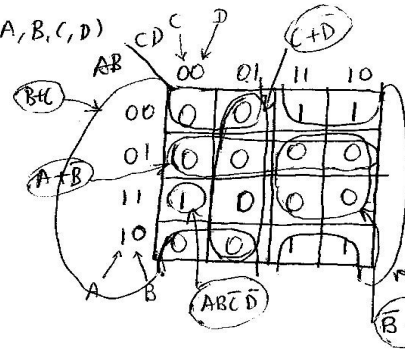
RAPPRESENTAZIONE DI FUNZIONI BOOLEANE MEDIANTE TABELLE DI VERITA' E MAPPE DI KARNAUGH - FORME CANONICHE - MINIMIZZAZIONE

1° ESEMPIO

Se la funzione booleana F è rappresentata mediante la tabella di verità, si può ottenere la sua rappresentazione, alternativamente, mediante la mappa di Karnaugh, marcando in ogni cella, individuata dai valori delle variabili indipendenti, il valore di $F(0,1)$

Tabella di verità $F(A,B,C,D)$

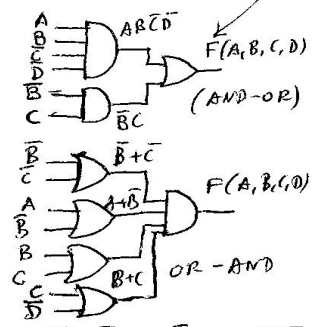
A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0



FORME MINIMIZZATE

$$F(A,B,C,D) = \overline{B}C + A\overline{B}C\overline{D}$$

$$F(A,B,C,D) = (\overline{B} + \overline{C})(A + \overline{B})(B + C)(C + \overline{D})$$



FORME CANONICHE DI $F(A,B,C,D)$

forma canonica disgiunta:

$$F(A,B,C,D) = \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + A\overline{B}C\overline{D} + A\overline{B}CD + A\overline{B}C\overline{D}$$

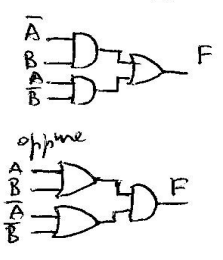
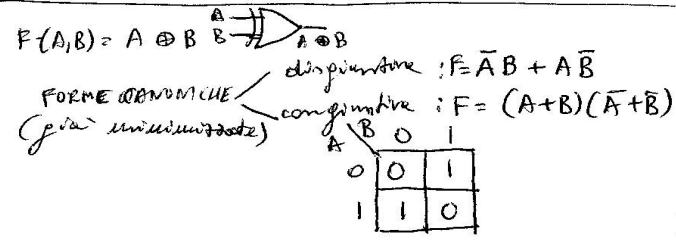
forma canonica congiunta

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

2° esempio

PORTA EXOR

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0



Minimizzazione di funzioni booleane con il metodo delle mappe di Karnaugh - ESEMPIO -

1)

Tabella di verità

A	B	C	F(A,B,C)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Forme canoniche:

$F(A,B,C) = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

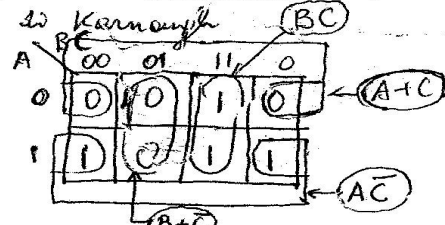
$F(A,B,C) = (A+B+C)(A+B+\bar{C})(A+\bar{B}+C) \cdot (A+B+\bar{C})$

Forme minimizzate:

$F(A,B,C) = A\bar{C} + BC$

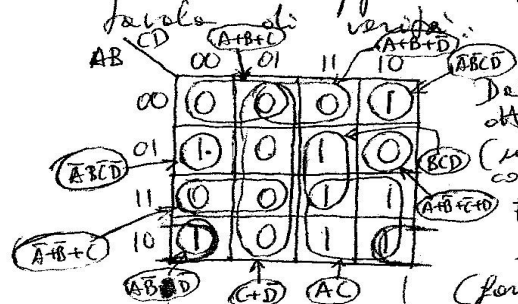
$F(A,B,C) = (A+C)(B+\bar{C})$

→ mappa di Karnaugh



Se si considerano le caselle marcate con 0, si associano agli 0 le variabili dirette (A,B,C) ed agli 1 le variabili negate ($\bar{A}, \bar{B}, \bar{C}$); se si considerano invece le caselle marcate con 1, si associano agli 0 le variabili negate ($\bar{A}, \bar{B}, \bar{C}$) ed agli 1 le variabili dirette (A,B,C). Si considerano infine le aree più estese possibili che possano ricoprire tutto gli 1 (o tutti gli 0) e si trova la somma (o il prodotto) dei termini corrispondenti.

2) Si suppone che la funzione $F(A,B,C,D)$ venga assegnata con la mappa di Karnaugh in sostituzione della



Dalla mappa di Karnaugh si possono ottenere subito le forme canoniche (non minimizzate) leggendo i mintermi corrispondenti agli 1 ed i maxtermi corrispondenti agli 0.

$F(A,B,C,D) = \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}B\bar{C}\bar{D}$

(forme canoniche disgiuntive)

$F(A,B,C,D) = (A+B+C+D)(A+B+C+\bar{D})(A+B+\bar{C}+D)(A+\bar{B}+C+D)(A+\bar{B}+\bar{C}+D) \cdot (\bar{A}+\bar{B}+C+D)(\bar{A}+\bar{B}+C+\bar{D})(\bar{A}+\bar{B}+C+D)$

forma canonica congiuntiva

Forme minimizzate:

$F(A,B,C,D) = \bar{A}\bar{B}C\bar{D} + BCD + AC + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D}$

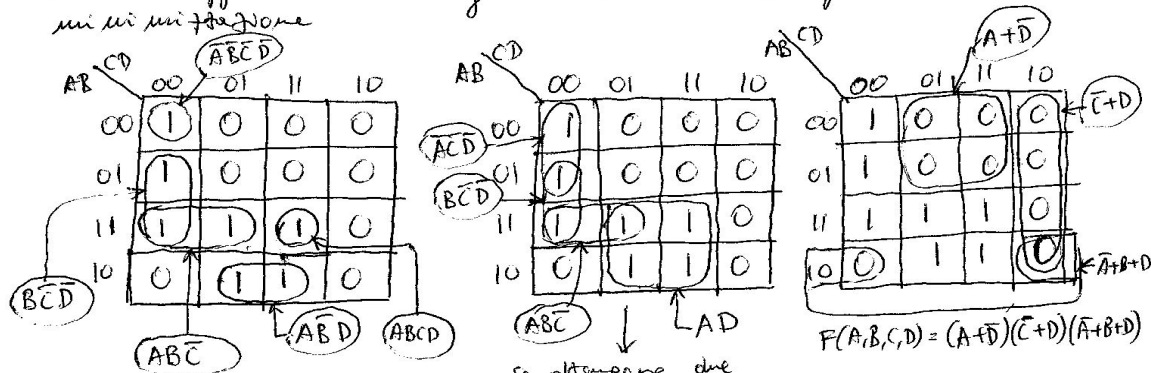
(disgiuntiva e congiuntiva)

$F(A,B,C,D) = (C+\bar{D})(A+B+C)(A+B+\bar{D})(A+\bar{B}+\bar{C}+D) \cdot (\bar{A}+\bar{B}+C)$

- 3) Si suppone che la funzione $F(A,B,C,D)$ venga assegnata attraverso un'espressione booleana.

$$F(A,B,C,D) = \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C} + A\bar{B}D + B\bar{C}\bar{D} + ABCD$$

In questo caso bisogna prima rappresentare i singoli termini sulle mappe di Karnaugh e successivamente procedere alla minimizzazione.



Si ottengono due soluzioni (forme minimizzate disgiunte)

$$F(A,B,C,D) = \bar{A}\bar{C}\bar{D} + \bar{B}\bar{C}\bar{D} + AD$$

$$F(A,B,C,D) = \bar{A}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C} + AD$$

PROGETTO DI UNA RETE LOGICA COMBINATORIA

11

Il sistema di sicurezza di un impianto industriale deve entrare in funzione se le elettrovalvole V_1 e V_2 sono aperte ed il segnale S_1 è al livello alto, oppure se l'elettrovalvola V_2 è aperta ed il segnale S_1 è al livello basso, oppure se l'elettrovalvola V_1 è chiusa ed i segnali S_1 ed S_2 sono al livello basso, oppure se l'elettrovalvola V_1 è chiusa, il segnale S_1 è al livello alto ed il segnale S_2 è al livello basso.

Y $\begin{cases} 1 & \text{sistema di sicurezza inserito} \\ 0 & \text{sistema di sicurezza disinserito} \end{cases}$

V_1, V_2 $\begin{cases} 0 & \text{elettrovalvola chiusa} \\ 1 & \text{" aperta} \end{cases}$

S_1, S_2 $\begin{cases} 0 & \text{segnale al livello basso} \\ 1 & \text{segnale al livello alto} \end{cases}$

Per dedurre la funzione booleana $Y(V_1, V_2, S_1, S_2)$ si considerano le variabili V_1 e V_2 in forma diretta o negata ($\bar{V}_1, \bar{V}_2$) a seconda che le corrispondenti elettrovalvole siano rispettivamente aperte (V_1, V_2) o chiuse ($\bar{V}_1, \bar{V}_2$). Analogamente si considerano le variabili S_1 ed S_2 in forma diretta o negata ($\bar{S}_1, \bar{S}_2$) a seconda che i corrispondenti segnali siano rispettivamente al livello alto (S_1, S_2) o al livello basso ($\bar{S}_1, \bar{S}_2$).

V_1, V_2, S_1 ed S_2 possono essere segnali elettrici o di pressione (nel caso in cui si considerano controlli idraulici o pneumatici)

$$Y(V_1, V_2, S_1, S_2) = V_1 V_2 S_1 + V_2 \bar{S}_1 + \bar{V}_1 \bar{S}_1 \bar{S}_2 + \bar{V}_1 S_1 \bar{S}_2$$

MAPPA DI KARNAUGH

$V_1 \backslash V_2 \backslash S_1 S_2$	00	01	11	10
$\bar{V}_1 \bar{S}_2$	1	0	0	1
$\bar{V}_1 S_2$	1	1	0	1
$V_2 \bar{S}_1$	1	1	1	1
$V_2 S_1$	0	0	0	0

Gruppi circoscritti: $\bar{V}_1 \bar{S}_2$, $\bar{V}_1 S_2$, $V_2 \bar{S}_1$, $V_2 S_1$

$V_1 \backslash V_2 \backslash S_1 S_2$	00	01	11	10
$\bar{V}_1 \bar{S}_2$	1	0	0	1
$\bar{V}_1 S_2$	1	1	0	1
$V_2 \bar{S}_1$	1	1	1	1
$V_2 S_1$	0	0	0	0

Gruppi circoscritti: $\bar{V}_1 \bar{S}_2$, $\bar{V}_1 S_2$, $V_2 \bar{S}_1$, $V_2 S_1$

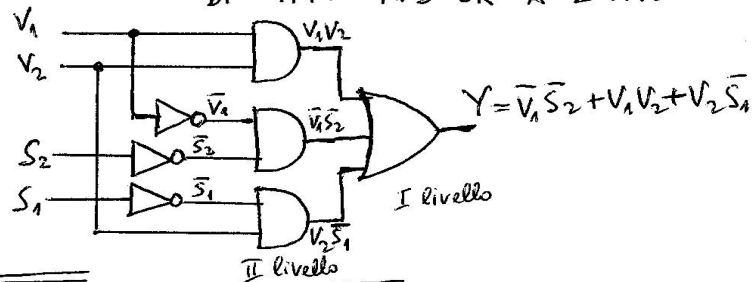
$Y = \bar{V}_1 \bar{S}_2 + V_1 V_2 + V_2 \bar{S}_1$
(MINIMIZZATA)
IN FORMA DISGIUNTIVA

VERIFICA

$$Y = \bar{V}_1 \bar{S}_2 + V_1 V_2 + V_2 \bar{S}_1$$

V_1	V_2	S_1	S_2	Y	$\bar{V}_1$	$\bar{V}_2$	$\bar{S}_1$	$\bar{S}_2$	$\bar{V}_1 \bar{S}_2$	$V_1 V_2$	$V_2 \bar{S}_1$	Y
0	0	0	0	1	1	1	1	1	1	0	0	1
0	0	0	1	0	1	1	1	0	0	0	0	0
0	0	1	0	1	1	1	0	1	1	0	0	1
0	0	1	1	0	1	1	0	0	0	0	0	0
0	1	0	0	1	1	0	1	1	1	0	1	1
0	1	0	1	1	1	0	1	0	1	0	1	1
0	1	1	0	1	1	0	0	1	1	0	0	1
0	1	1	1	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	1	1	1	0	0	0	0
1	0	0	1	0	0	1	1	0	0	0	0	0
1	0	1	0	0	0	1	0	1	0	0	0	0
1	0	1	1	0	0	1	0	0	0	0	0	0
1	1	0	0	1	0	0	1	1	0	1	1	1
1	1	0	1	1	0	0	1	0	0	1	1	1
1	1	1	0	1	0	0	0	1	0	1	0	1
1	1	1	1	1	0	0	0	0	0	1	0	1

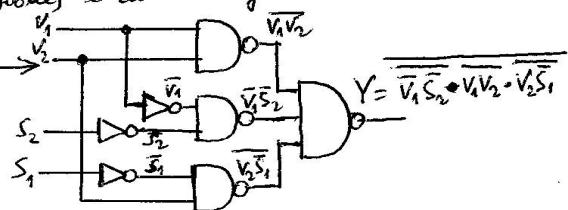
RETE LOGICA MINIMIZZATA
DI TIPO AND-OR A 2 LIVELLI

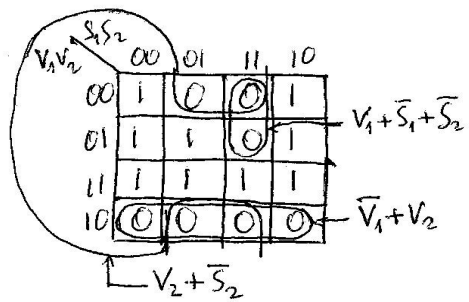


$$Y = \bar{V}_1 \bar{S}_2 + V_1 V_2 + V_2 \bar{S}_1 = \bar{V}_1 \bar{S}_2 + V_1 V_2 + V_2 \bar{S}_1$$

si applicano i teoremi di involuzione (doppia negazione) e di De Morgan

RETE LOGICA MINIMIZZATA
DI TIPO NAND-NAND A 2 LIVELLI





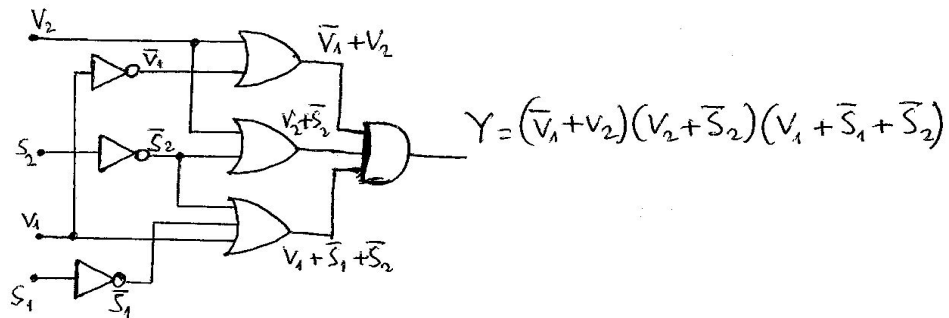
$$Y = (\bar{V}_1 + V_2)(V_2 + \bar{S}_2)(V_1 + \bar{S}_1 + \bar{S}_2)$$

(MINIMIZZATA)
IN FORMA CONGIUNTIVA

VERIFICA

V_1	V_2	S_1	S_2	Y	$\bar{V}_1$	$\bar{V}_2$	$\bar{S}_1$	$\bar{S}_2$	$\bar{V}_1 + V_2$	$V_2 + \bar{S}_2$	$V_1 + \bar{S}_1 + \bar{S}_2$	Y
0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	0	1	1	1	0	1	0	1	0
0	0	1	0	1	1	1	0	1	1	1	1	1
0	0	1	1	0	1	1	0	0	1	0	0	0
0	1	0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	1	0	1	0	1	1	1	1
0	1	1	0	1	1	0	0	1	1	1	1	1
0	1	1	1	0	1	0	0	0	1	1	0	0
1	0	0	0	0	0	1	1	1	0	1	1	0
1	0	0	1	0	0	1	1	0	0	0	1	0
1	0	1	0	0	0	1	0	1	0	1	1	0
1	0	1	1	0	0	1	0	0	0	0	1	0
1	1	0	0	1	0	0	1	1	1	1	1	1
1	1	0	1	1	0	0	1	0	1	1	1	1
1	1	1	0	1	0	0	0	1	1	1	1	1
1	1	1	1	1	0	0	0	0	1	1	1	1

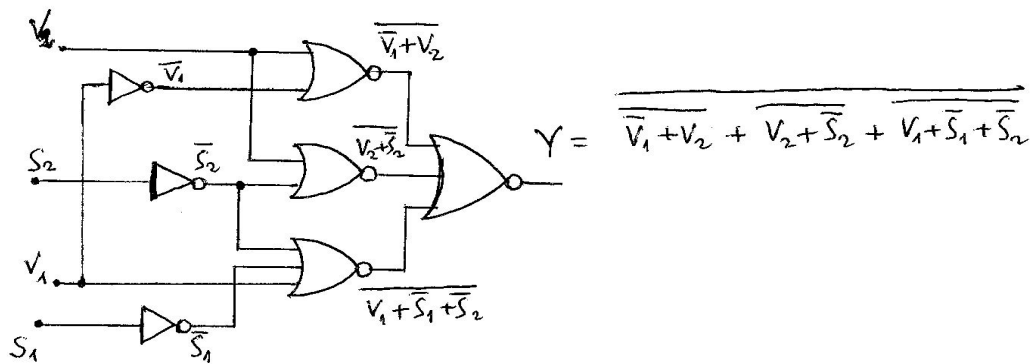
RETE LOGICA MINIMIZZATA
DI TIPO OR-AND A 2 LIVELLI



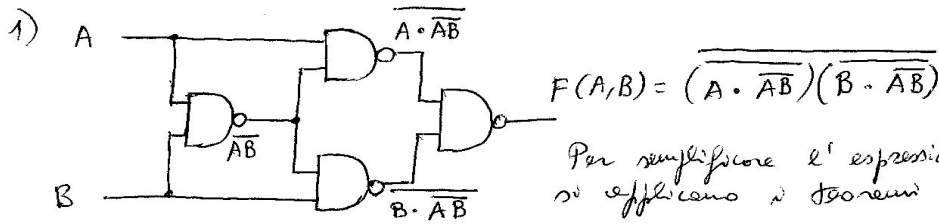
$$Y = (\overline{V_1} + V_2)(V_2 + \overline{S_2})(V_1 + \overline{S_1} + \overline{S_2}) = \overline{\overline{V_1} + V_2 + V_2 + \overline{S_2} + V_1 + \overline{S_1} + \overline{S_2}}$$

Si applicano i teoremi di involuzione e di De Morgan

RETE LOGICA MINIMIZZATA
DI TIPO NOR-NOR A 2 LIVELLI



Si tratta di ricavare la funzione booleana F conoscendo lo schema della rete logica che la realizza.



Per semplificare l'espressione di F si applicano i teoremi di De Morgan:

$$F(A,B) = \overline{Y \cdot Z} = \overline{Y} + \overline{Z} = \overline{A \cdot \overline{AB}} + \overline{B \cdot \overline{AB}} = A \cdot \overline{\overline{AB}} + B \cdot \overline{\overline{AB}} =$$

Si pone: $Y = A \cdot \overline{AB}$

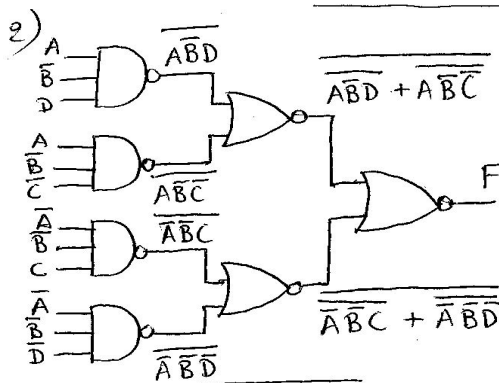
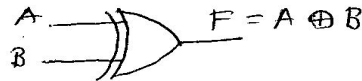
$Z = B \cdot \overline{AB}$

$$= A(\overline{\overline{AB}}) + B(\overline{\overline{AB}}) =$$

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

$$= A \oplus B$$

Pertanto la rete logica realizzata equivale alla semplice porta ~~EXOR~~



Si semplifica l'espressione di F applicando i teoremi di idempotenza e di De Morgan.

$$F = \overline{(A\overline{B} + AC) (\overline{B}C + C\overline{D})} =$$

$$= \overline{X + Y} = \overline{X} \cdot \overline{Y} =$$

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

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

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

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

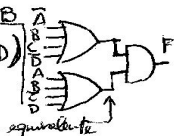
II soluzione

rete logica equivalente

$$F = \overline{A\overline{B} + AC} \cdot \overline{\overline{B}C + C\overline{D}} = \overline{A\overline{B} + AC} \cdot \overline{\overline{B}C + C\overline{D}} =$$

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

I soluzione



SEMISOMMATORE (HALF ADDER)

Il semisommatore esegue la somma di 2 cifre binarie senza tenere conto del riporto (carry) proveniente dalle cifre di ordine inferiore, e genera il riporto per le cifre di ordine superiore.

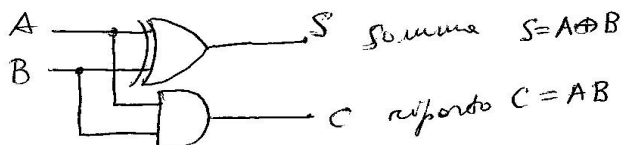
Tabella di verità

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

cifre binarie

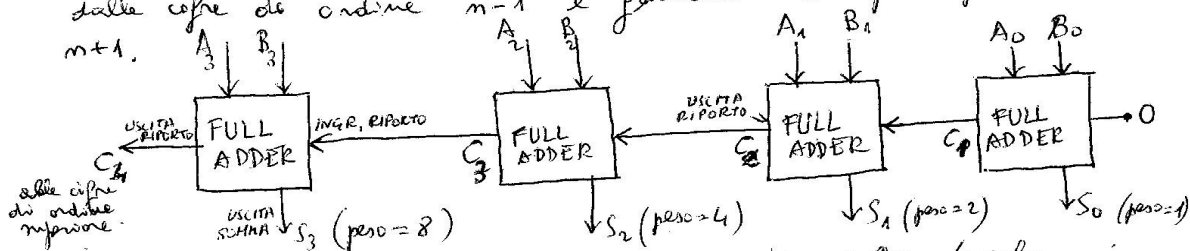
Schema logico del semisommatore

La somma $S = \bar{A}B + A\bar{B}$ si ottiene con una porta EXOR; il riporto C per l'ordine superiore si ottiene con una porta AND.



SOMMATORE COMPLETO (FULL ADDER)

Il sommatore completo esegue la somma di 2 cifre binarie (relative all'ordine n) tenendo conto del riporto proveniente dalle cifre di ordine $n-1$ e generando il riporto per l'ordine $n+1$.



Schema logico di sommatore per numeri binari a 4 cifre

L'ingresso di riporto delle Addio di destra viene automaticamente al livello logico 0.

Tabella di verità

A_n	B_n	C_n	S_n	C_{n+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$\begin{aligned}
 S_n &= \bar{A}_n \bar{B}_n C_n + \bar{A}_n B_n \bar{C}_n + A_n \bar{B}_n C_n + A_n B_n \bar{C}_n = \\
 \text{forma canonica} &= (\bar{A}_n B_n + A_n \bar{B}_n) \bar{C}_n + (A_n B_n + \bar{A}_n \bar{B}_n) C_n = \\
 &= (A_n \oplus B_n) \bar{C}_n + (A_n \oplus B_n) C_n = A_n \oplus B_n \oplus C_n \\
 C_{n+1} &= \bar{A}_n B_n C_n + A_n \bar{B}_n C_n + A_n B_n \bar{C}_n + A_n B_n C_n = \\
 \text{forma canonica} &= A_n B_n (\bar{C}_n + C_n) + (\bar{A}_n B_n + A_n \bar{B}_n) C_n = \\
 &= A_n B_n + (A_n \oplus B_n) C_n
 \end{aligned}$$

Il comparatore di uguaglianza a 1 bit genera un livello logico alto soltanto se le variabili d'ingresso A e B assumono lo stesso valore.

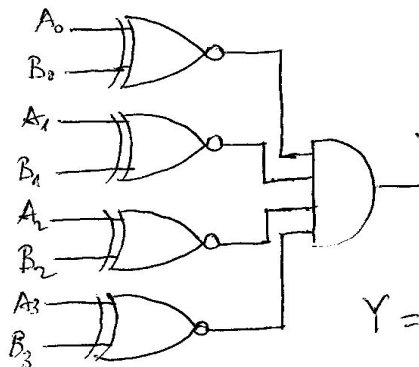
Tabella di verità

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

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

Il comparatore di uguaglianza a n bit si ottiene collegando in AND le uscite di n comparatori di uguaglianza a 1 bit.

Schema logico di comparatore di uguaglianza per numeri con 4 bit (A_3, A_2, A_1, A_0) (B_3, B_2, B_1, B_0)



$$Y = 1 \text{ solo se } \begin{matrix} A_0 = B_0 \\ A_1 = B_1 \\ A_2 = B_2 \\ A_3 = B_3 \end{matrix}$$

$$Y = (A_0 \odot B_0)(A_1 \odot B_1)(A_2 \odot B_2)(A_3 \odot B_3)$$

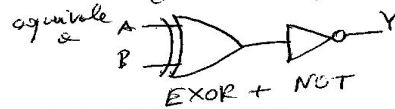
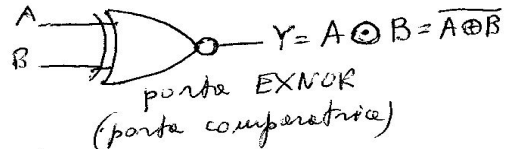
Se si considera la tabella di verità della porta EXOR,

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

Si vede che la funzione $Y = \bar{A}\bar{B} + AB$ coincide con la negazione della funzione EXOR;

$$\text{risultato } Y = \overline{A \oplus B}$$

Si usa anche l'operatore coincidente $\odot$ che corrisponde al NOR esclusivo (EXNOR)



CIRCUITI DI CODIFICA E DI DECODIFICA

TABELLA DI VERITÀ E

SCHEMA DI CODIFICATORE (ENCODER) DECIMALE/BCD

↑
Questa cifra decimale viene
rappresentata in codice
BCD (Binary Code Decimal)
con 4 bit

Tabella di verità

INGRESSI DECIMALI									USCITE BINARIE			
X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	Y_3	Y_2	Y_1	Y_0
1	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	0	0	1	0	0	0

L'uscita Y_0 deve assumere il livello alto soltanto
attivando le linee d'ingresso corrispondenti alle
cifre 1, 3, 5, 7, 9 ; pertanto $Y_0 = X_1 + X_3 + X_5 + X_7 + X_9$
Analogamente si ha:

$$Y_1 = X_2 + X_3 + X_8 + X_7$$

$$Y_2 = X_4 + X_5 + X_6 + X_7$$

$$Y_3 = X_8 + X_9$$

Se nessuna delle linee 1...9
si trova al livello alto, le
uscite Y_0, Y_1, Y_2, Y_3 si trovano al livello basso (corrispondente alla cifra
decimale 0)

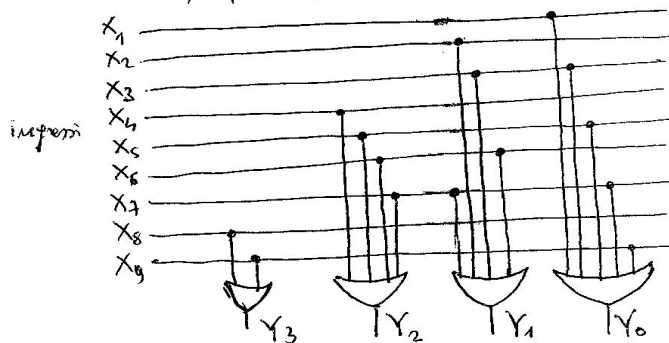


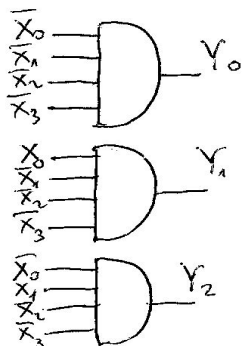
TABELLA DI VERITA' E SCHEMA DI DECODIFICATORE BCD/DECIMALE

TABELLA DI VERITA'

INGRESSI IN BCD				USCITE DECIMALI									
X_3	X_2	X_1	X_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

Bisogna esprimere 10 funzioni booleane aventi ciascuna un solo mintermine attivo che corrisponde alla combinazione binaria da decodificare. $(Y_0, \dots, Y_9)$

$$\text{Es. } Y_0 = \bar{X}_0 \bar{X}_1 \bar{X}_2 \bar{X}_3, \quad Y_1 = X_0 \bar{X}_1 \bar{X}_2 \bar{X}_3, \quad Y_2 = \bar{X}_0 X_1 \bar{X}_2 \bar{X}_3$$



Occorrono pertanto 10 porte AND a 4 ingressi e 4 porte NOT (per ottenere $\bar{X}_0, \bar{X}_1, \bar{X}_2, \bar{X}_3$)

Differenza tra reti logiche combinatorie e reti logiche sequenziali

Le reti logiche combinatorie sono caratterizzate dal fatto che le loro uscite in un determinato istante dipendono non soltanto dai livelli logici assunti dalle variabili d'ingresso (variabili indipendenti) nello stesso istante, invece nelle reti logiche sequenziali, le contemporaneamente non soltanto porte logiche ma uno o più elementi di memoria (bistabili o flip-flop), le uscite dipendono non soltanto dai valori attuali assunti dalle variabili d'ingresso, ma anche dallo stato della rete, cioè dai valori di un insieme di variabili di stato, che generalmente coincidono con le variabili d'uscita e che rappresentano la "storia" della rete logica sequenziale. Pertanto una rete logica sequenziale è costituita da un insieme di porte logiche e da 1 o più flip-flop, che sono gli elementi fondamentali (elementi di memoria) destinati a memorizzare i valori delle uscite in un determinato istante, valori che determinano il comportamento della rete negli istanti successivi.

PRINCIPALI TIPI DI FLIP-FLOP (ELEMENTI BISTABILI)

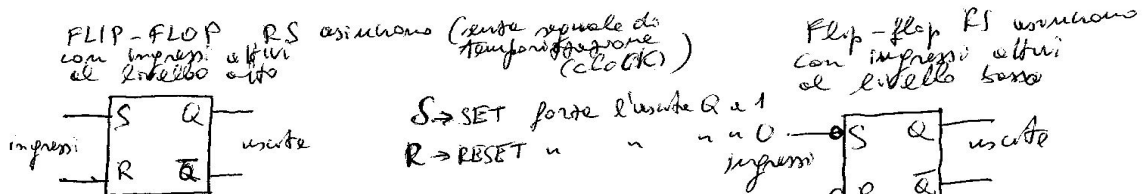


Tabelle di verità

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	—

(Si suppone che per variare i livelli dopo l'ingresso in determinati istanti, t_{n-1}, t_n, t_{n+1} , numerabili)

il flip-flop all'istante t_{n+1} assume lo stato Q_n relativo all'istante precedente (t_n)

← combinazione non usata

Tabelle di verità

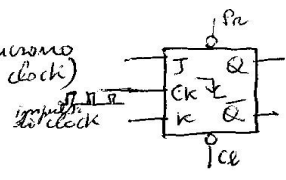
S	R	Q_{n+1}
0	0	1
0	1	0
1	1	Q_n

combinazione non usata

FLIP-FLOP JK Simbolo (con segnale di clock)

Tabella di verità

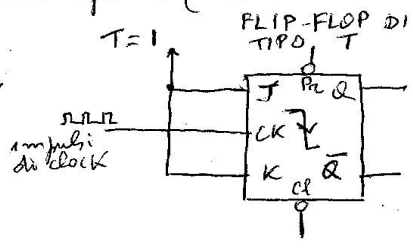
J	K	Q _{n+1}
0	0	Q _n
0	1	0
1	0	1
1	1	$\bar{Q}_n$



gli ingressi Pe e Ce (Preset e clear) predispongono rispettivamente a 1 e a 0 l'uscita Q, indipendentemente dal segnale di clock e dai valori presenti da J e K.

Vediamo che il flip-flop JK non possiede combinazioni vietate.

Se, in particolare, gli ingressi J e K vengono costantemente mantenuti al livello alto, il flip-flop commuta ad ogni impulso di clock assumendo uno stato Q_{n+1} complementare a quello precedente; ~~per questo~~ si trasforma in un flip-flop di tipo T (TOGGLE - commutazione).



T = J = K = 1
Tabella di verità

T	Q _{n+1}
0	Q _n
1	$\bar{Q}_n$

Se l'ingresso T viene mantenuto al livello basso, il flip-flop non commuta. ($Q_{n+1} = Q_n$)

FLIP-FLOP DI TIPO D (DATA LATCH)

Si ottiene dal flip-flop JK aggiungendo una porta NOT.

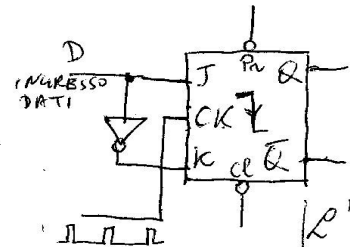


Tabella di Verità

D	Q _{n+1}
0	0
1	1

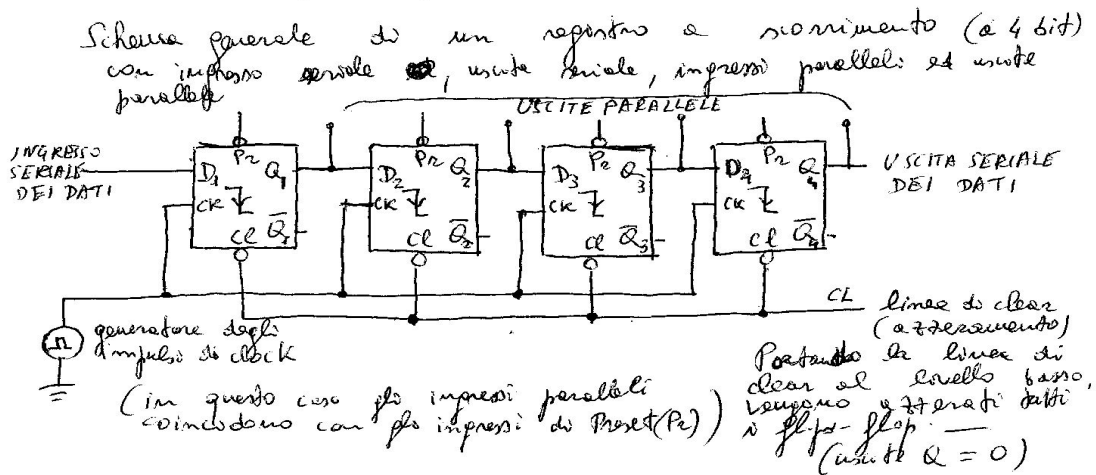
$$Q_{n+1} = D_n$$

(PRESET) Pe → predisporre l'uscita Q a 1 (attivo al livello basso)
(CLEAR) Ce → predisporre l'uscita Q a 0.

L'uscita Q assume il livello logico presente all'ingresso D all'arrivo dell'impulso di clock, e lo memorizza fino all'arrivo del successivo impulso di clock che assume lo stato del flip-flop in funzione dello stato assunto dall'ingresso D.

Impiego: memoria dati nei sistemi digitali.

I registri sono gli organi fondamentali di memoria dei sistemi digitali di elaborazione e controllo. Un registro è costituito da più flip-flop (di tipo SR, JK o D) collegati in cascata, in modo che l'informazione inizialmente immagazzinata negli n flip-flop possa scorrere da un bit verso destra o verso sinistra per ogni impulso di clock applicato.



In generale i registri sono dotati anche di uscite parallele (una per ogni flip-flop) e di ingressi paralleli (1 per ogni flip-flop) al fine di poter immagazzinare ed estrarre l'informazione in tutti i modi possibili ed assicurare con una grande flessibilità d'impiego.

modalità di immissione ed estrazione dei dati

- ingresso seriale ed uscita seriale S.I.S.O.
- ingresso seriale ed uscita in parallelo S.I.P.O
- ingresso in parallelo ed uscita seriale P.I.S.O
- ingresso in parallelo ed uscita in parallelo P.I.P.O

Si parla di uscita in parallelo ed ingresso in parallelo quando i 4 bit del dato vengono estratti o immessi simultaneamente. Si parla invece di uscita seriale ed ingresso seriale quando i singoli bit del dato vengono estratti o immessi ad ogni impulso di clock.

REGISTRI A SCORRIMENTO

23

$P_1, IP_1, IP_2, IP_3, IP_4$
(INGRESSI PARALLELI)
 UP_1, UP_2, UP_3, UP_4
(USCITE PARALLELE)

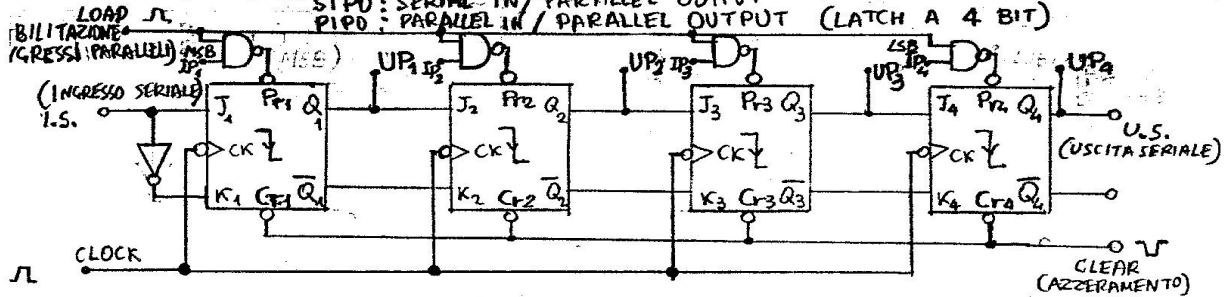
REGISTRO A SCORRIMENTO DI TIPO UNIVERSALE

SISO: SERIAL IN / SERIAL OUTPUT

PISO: PARALLEL IN / SERIAL OUTPUT

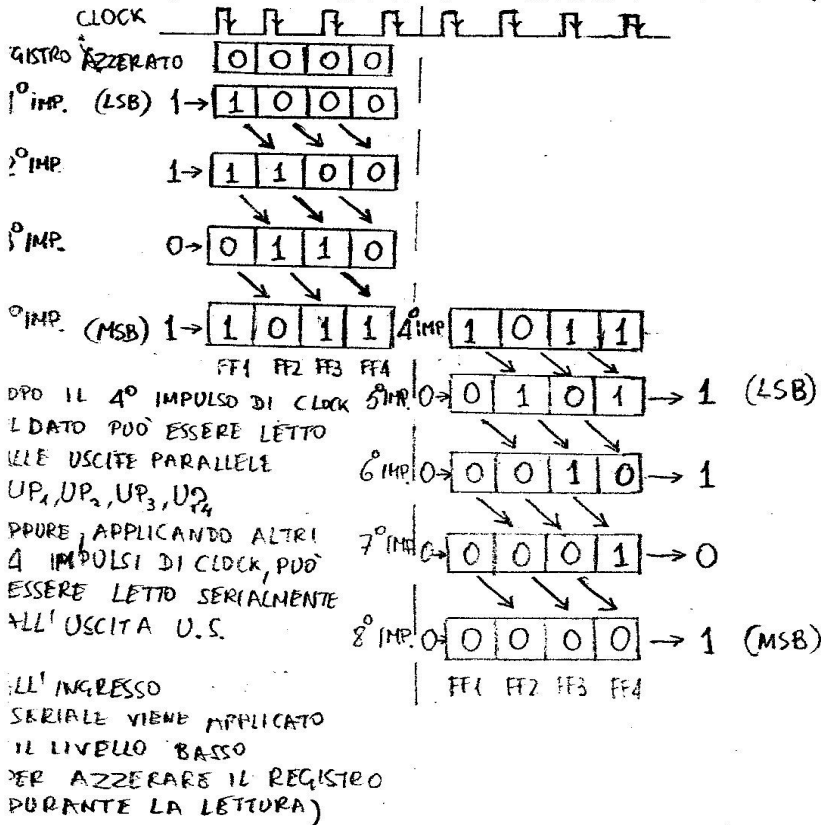
SIPD: SERIAL IN / PARALLEL OUTPUT

PIPD: PARALLEL IN / PARALLEL OUTPUT (LATCH A 4 BIT)



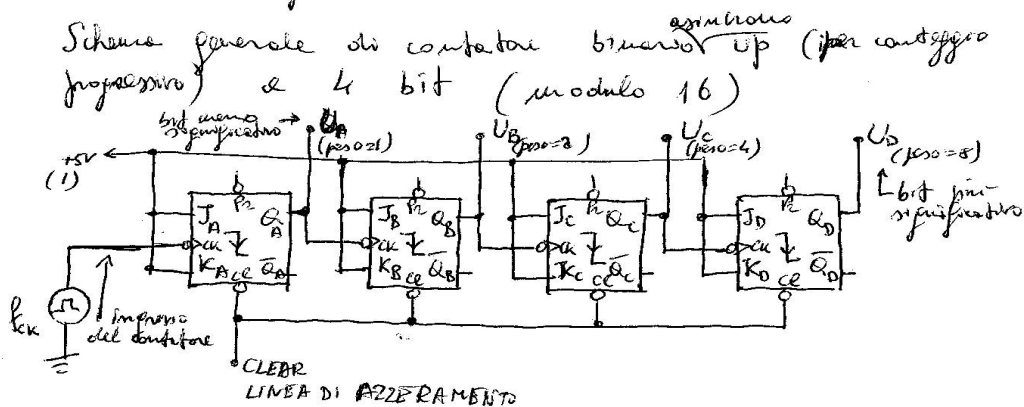
I FLIP-FLOP SONO DEL TIPO MASTER-SLAVE

ESEMPIO: IL DATO 1011 VIENE CARICATO SERIALMENTE CON I PRIMI 4 IMPULSI DI CLOCK E SUCCESSIVAMENTE LETTO SERIALMENTE MEDIANTE ALTRI 4 IMPULSI DI CLOCK



I contatori ^{binari} sono circuiti logici sequenziali utilizzati qualora si desideri effettuare il conteggio di impulsi (espresso in codice binario) attraverso gli stati assunti dalle uscite parallele, o qualora sia necessario dividere per una potenza di 2 la frequenza del segnale di clock applicato all'ingresso del dispositivo (impiego del contatore come divisore di frequenza).

Un contatore binario asincrono è costituito da 2 o più flip-flop di tipo T (o JK usati come flip-flop T) collegati in cascata, al fine di fornire attraverso le uscite (Q_i) dei flip-flop lo stato del conteggio degli impulsi di clock che arrivano all'ingresso del primo flip-flop.



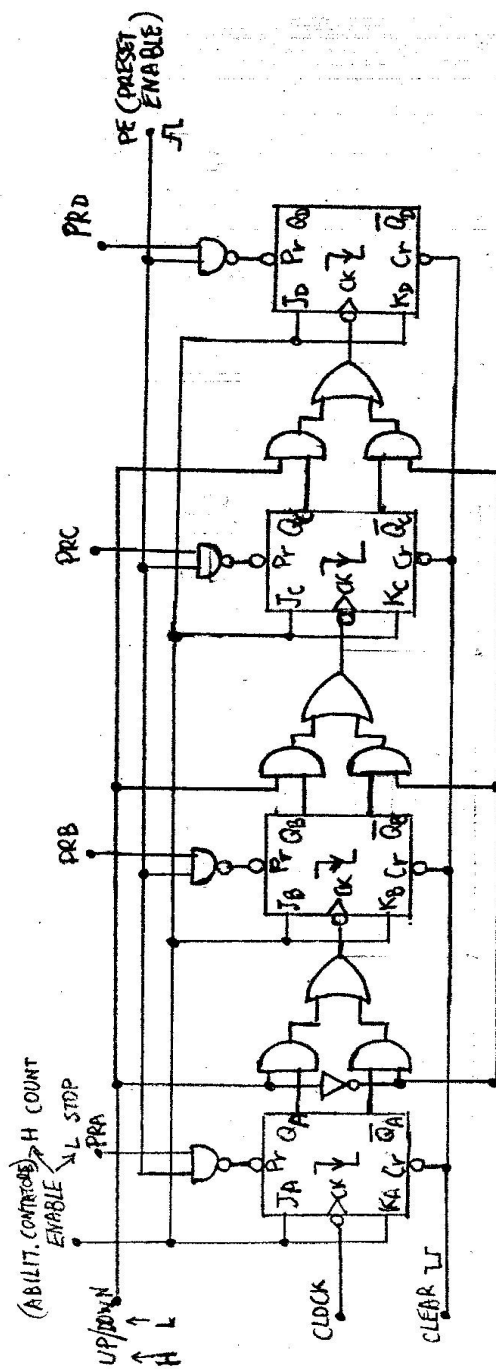
Il modulo del contatore è dato dal numero di impulsi che bisogna applicare all'ingresso affinché il contatore (inizialmente azzerato) si ripresenti nello stato iniziale (accensione di nuovo del contatore) dopo aver effettuato il massimo conteggio.

La capacità del contatore è invece $C = M - 1 = 2^n - 1$, cioè coincide con la più elevata combinazione binaria presente alle uscite prima del riciclo.

Se $n = 4$ $M = 16$ $C = 15 \rightarrow (1111)_2$

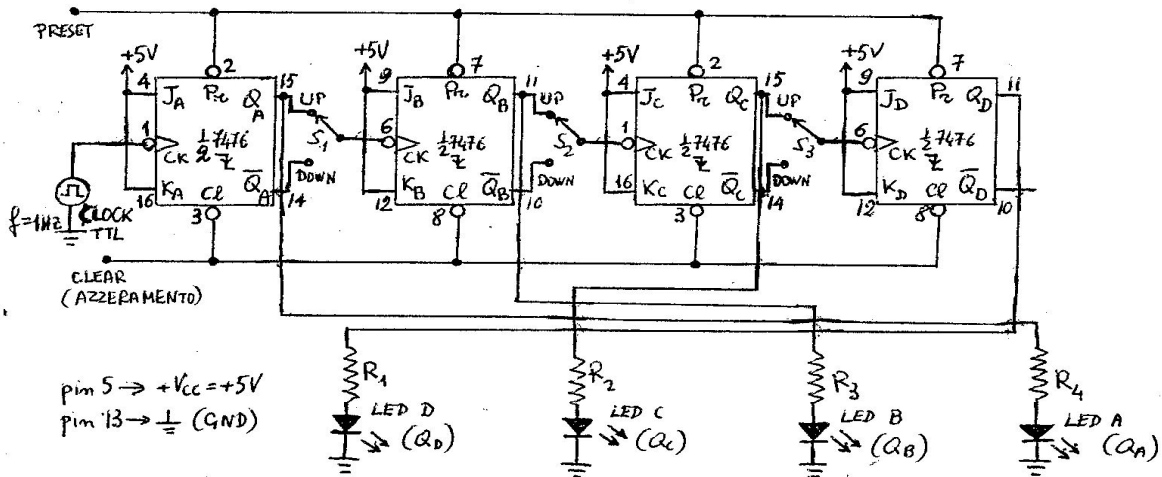
Ogni flip-flop del contatore divide per 2 la ^{in binario} frequenza del segnale applicato all'ingresso CK; pertanto il segnale rettangolare presente all'uscita Q_0 ha una frequenza pari a $f_{CK}/16$. (divisore di frequenza per 16).

CONTATORE BINARIO ASINCRONO UP/DOWN MODULO 16 PRESETTABILE



Contatore binario asincrono up-down, modulo 16, realizzato con flip-flop JK master-slave.

26



I simboli $\uparrow$ e $\downarrow$ indicano che i flip-flop commutano in corrispondenza del fronte di discesa dell'impulso di clock (clock di transizione attivo basso)

$$R_1 = R_2 = R_3 = R_4 = 180\Omega$$

Per predisporre le uscite Q_A , Q_B , Q_C e Q_D al livello logico 1 (livello alto), occorre applicare il livello logico 0 agli ingressi di preset (ingressi di preset attivi bassi).

Per azzerare il contatore occorre applicare il livello logico 0 agli ingressi di clear (ingressi di clear attivi bassi).

Per ottenere il conteggio in avanti (up) i deviatori S_1, S_2, S_3 devono essere commutati nella posizione up.

Per ottenere il conteggio all'indietro (down) i deviatori S_1, S_2, S_3 devono essere commutati nella posizione down.

Sequenza delle commutazioni e forme d'onda relative al contatore ^{binario} asincrono up (contatore ^{contatore} in avanti), modulo 16.

27

numero impulsi in clock	N (equiv. decimale)	Q _D	Q _C	Q _B	Q _A
1	0	0	0	0	0
2	1	0	0	0	1
3	2	0	0	1	0
4	3	0	0	1	1
5	4	0	1	0	0
6	5	0	1	0	1
7	6	0	1	1	0
8	7	0	1	1	1
9	8	1	0	0	0
10	9	1	0	0	1
11	10	1	0	1	0
12	11	1	0	1	1
13	12	1	1	0	0
14	13	1	1	0	1
15	14	1	1	1	0
16	15	1	1	1	1
	0	0	0	0	0

← contatore resettato

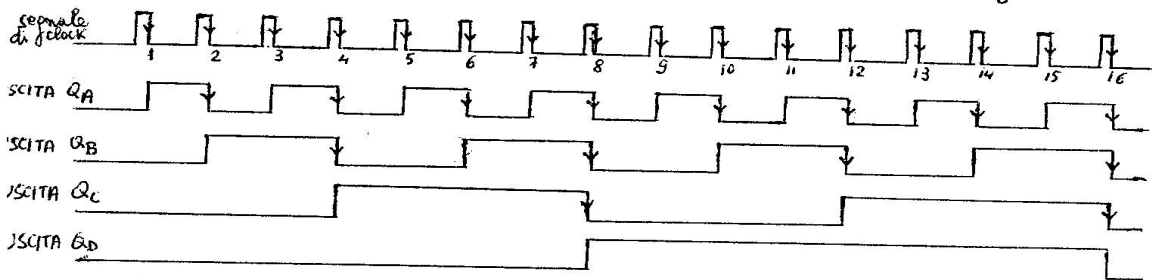
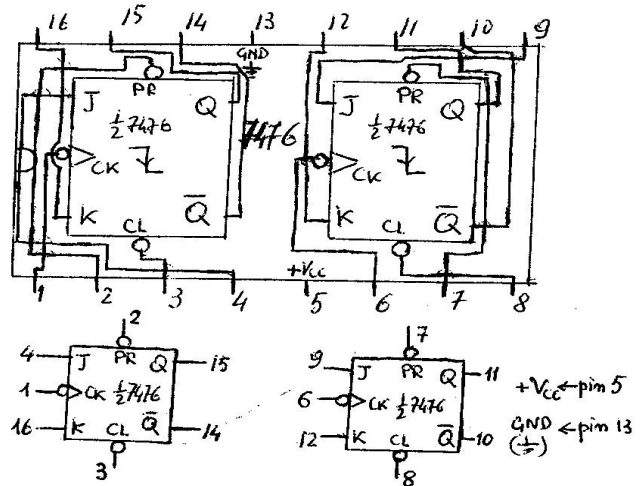


Tabella della verità ed equazione di stato del flip-flop JK

J	K	Q _{n+1}
0	0	Q _n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

$$Q_{n+1} = J\overline{Q_n} + \overline{K}Q_n$$

Se agli ingressi J e K viene applicato costantemente il livello logico 1, il flip-flop JK si comporta come flip-flop di tipo T (Toggle) e commuta ad ogni impulso del clock (da 0 a 1 e da 1 a 0) ($Q_{n+1} = \overline{Q_n}$)

Sequenza delle commutazioni e forme d'onde relative
 al contatore ^{binario} ~~asincrono~~ ^{asincrono} down (conteggio all'indietro), modulo 16

28

numero impulsi di clock	N (equiv. decimale)	Q_D	Q_C	Q_B	Q_A
1	15	1	1	1	1
2	14	1	1	0	0
3	13	1	1	0	1
4	12	1	1	0	0
5	11	1	0	1	1
6	10	1	0	1	0
7	9	1	0	0	1
8	8	1	0	0	0
9	7	0	1	1	1
10	6	0	1	1	0
11	5	0	1	0	1
12	4	0	1	0	0
13	3	0	0	1	1
14	2	0	0	1	0
15	1	0	0	0	1
16	0	0	0	0	0
17	15	1	1	1	1

← contatore
prezettato

