

05EKL-Progetto di Circuiti Digitali

Tutore: Federico Quaglio

federico.quaglio@polito.it

011-564 4004 (4004)

Introduzione alle Reti Logiche

Sommario

- **Richiami di algebra booleana**
- Mappe di Karnaugh
- Coperture SP & PS
- Alee statiche e loro minimizzazione
- Circuiti sequenziali semplici
- Ulteriori Circuiti

Algebra Booleana

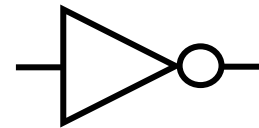
- Negazione o inversione: NOT

- $A=\{0,1\}$ $B=\text{NOT}(A)=\bar{A}=A'=\{1,0\}$

- Tabella di Verità:

A	B
0	1
1	0

- Simbolo:



(o)

Algebra Booleana

- Prodotto logico: AND;

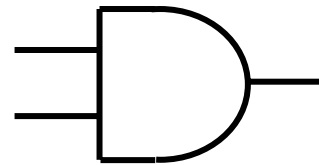
- $C=A*B=AB$

- Tavola di verità:

A	B	U
0	0	0
0	1	0
1	0	0
1	1	1

- $C = 1 \Leftrightarrow A=B=1$

Simbolo:



Algebra Booleana

- Somma logica: OR;

- $C=A+B$

- Tavola di Verità:

A	B	U
0	0	0
0	1	1
1	0	1
1	1	1

- $C = 0 \Leftrightarrow A=B=0$

Simbolo:



Algebra Booleana

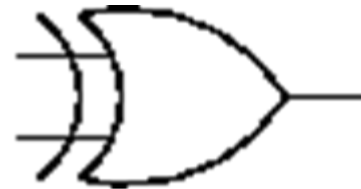
- OR Esclusivo: EXOR;

- $C = \overline{A} \cdot B + A \cdot \overline{B} = A \oplus B$

- Tavola di Verità:

A	B	U
0	0	0
0	1	1
1	0	1
1	1	0

- Simbolo:



- Comparatore di DISUGUAGLIANZA

Algebra Booleana

- NAND (AND Negata):

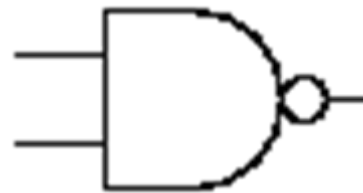
- $C = \overline{AB}$

- Tavola di verità:

A	B	U
0	0	1
0	1	1
1	0	1
1	1	0

- $C=0 \Leftrightarrow A=B=1$

Simbolo:



Algebra Booleana

- NOR (OR Negata):

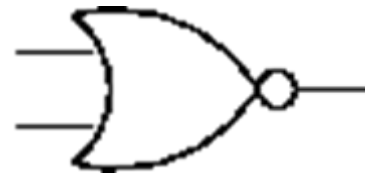
- $C = \overline{A + B}$

- Tavola di Verità:

A	B	U
0	0	1
0	1	0
1	0	0
1	1	0

- $C = 1 \Leftrightarrow A=B=0$

Simbolo:



Algebra Booleana

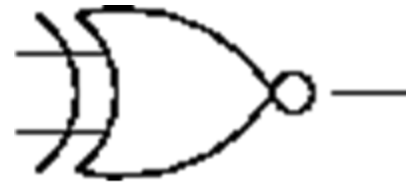
- EXNOR (EXOR Negata);

- $C = \overline{A} \cdot \overline{B} + A \cdot B = \overline{A \oplus B}$

- Tavola di Verità:

A	B	U
0	0	1
0	1	0
1	0	0
1	1	1

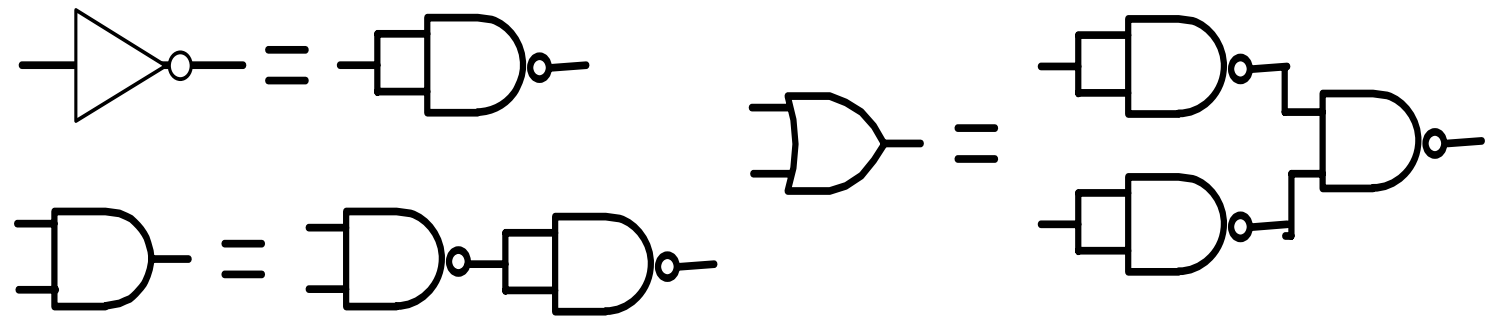
- Simbolo:



- Comparatore di UGUAGLIANZA

Algebra Booleana

- NAND, NOR SONO GATE UNIVERSALI
- Si possono ricavare tutte le altre funzioni logiche a partire da essi
- Es (con NAND):



Algebra Booleana

Proprietà fondamentali:

$$A*0=0; A+1=1; A*A=A; A+A=A$$

$$A*(NOT(A))=0; A+(NOT(A))=1$$

$$A+B=B+A; A*B=B*A$$

$$(A+B)+C=A+(B+C); (A*B)*C=A*(B*C)$$

$$A+A*B=A; A*(A+B)=A;$$

$$A+NOT(A)*B=A+B; A*(NOT(A)+B)=A*B$$

$$NOT(A+B)=NOT(A)*NOT(B)$$

$$NOT(A*B)=NOT(A)+NOT(B)$$

$$NOT(NOT(A))=A$$

Minimizzazione di Funzioni Logiche

- Somma di Prodotti (SP)
 - $f(A,B,C)=(A*B*C)+(A'BC)+ \text{etc...}$
 - In tabella di verità identifico i prodotti degli ingressi che hanno uscita '1' e li sommo
 - Ingresso a '1' \Leftrightarrow Considerato "dritto"
 - Ingresso a '0' \Leftrightarrow Considerato "negato"
- Prodotti di Somme (PS)
 - $f(A,B,C)=(A+B+C)(A'+B+C)\text{etc...}$
 - In tabella di verità identifico le somme che danno uscita '0' e le moltiplico
 - Ingresso a '0' \Leftrightarrow Considerato "dritto"
 - Ingresso a '1' \Leftrightarrow Considerato "negato"

Minimizzazione di Funzioni Logiche

- Ottengo rappresentazione in forma canonica
- Minimizzo funzione usando proprietà dell'algebra Booleana
 - Funzione minima

Minimizzazione di Funzioni Logiche

- Esempio (SP):

Tabella di verità

A	B	C	U
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$\begin{aligned}U &= I_1 + I_2 + I_3 + I_4 = \\ &= \bar{A}BC + \bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC\end{aligned}$$

$I_1 = \bar{A}BC$

$I_2 = \bar{A}\bar{B}C$

$I_3 = A\bar{B}\bar{C}$

$I_4 = ABC$

Minimizzazione di Funzioni Logiche

- Riduzione a espressione minima

$$\begin{aligned}U &= I_1 + I_2 + I_3 + I_4 = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC = \\&= BC(\bar{A} + A) + A\bar{B}C + AB\bar{C} = BC + A\bar{B}C + AB\bar{C} = \\&= B(C + A\bar{C}) + A\bar{B}C = B(C + A) + A\bar{B}C = \\&= AB + BC + A\bar{B}C = AB + C(B + A\bar{B}) = \\&= AB + C(B + A) = AB + AC + BC\end{aligned}$$

Sommario

- Richiami di algebra booleana
- **Mappe di Karnaugh**
- **Coperture SP & PS**
- Alee Statiche e loro minimizzazione
- Circuiti sequenziali semplici
- Ulteriori circuiti

Mappe di Karnaugh

- Servono a minimizzare espressioni logiche
- Copertura a Somma di Prodotti (SP)

$$f(A,B,C)=(A*B*C)+(A'BC)+ \text{etc...}$$

- Copertura a Prodotti di Somme (PS)

$$f(A,B,C)=(A+B+C)(A'+B+C)\text{etc...}$$

Mappe di Karnaugh

- Metodo grafico molto veloce
- Adatto per minimizzazione manuale
- Complessità esponenziale con numero di variabili di ingresso
- Al massimo fino a 5-6 ingressi

Mappe di Karnaugh

X

Y

	0	1
0	0	2
1	1	3

U=...

XY

W

	00	01	11	10
0	0	2	6	4
1	1	3	7	5

U=...

XY

WZ

	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

U=...

Mappe di Karnaugh

WZ \ XY		V=0				V=1			
		00	01	11	10	00	01	11	10
00	0	4	12	8	16	20	28	24	
01	1	5	13	9	17	21	29	25	
11	3	7	15	11	19	23	31	27	
10	2	6	14	10	18	22	30	26	

Da tabella di verità a Mappa di Karnaugh

- Tabella di verità

A	B	C	U
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- Mappa

		AB			
		00	01	11	10
C	0	0 0	0 2	1 6	0 4
	1	0 1	1 3	1 7	1 5

U

Regola per la copertura

- Iniziare a coprire gli '1' (o gli '0') non ancora coperti
- Massimizzare il "CUBO" includendo il maggior numero di '1' (o '0') in celle "Adiacenti"
- Celle sono Adiacenti \Leftrightarrow c'è la variazione di UN SOLO ingresso

Regola per la copertura

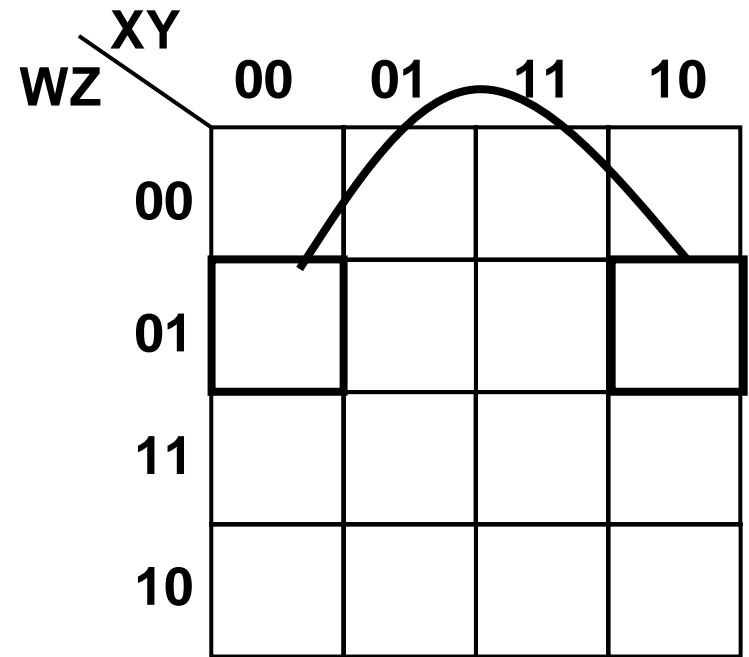
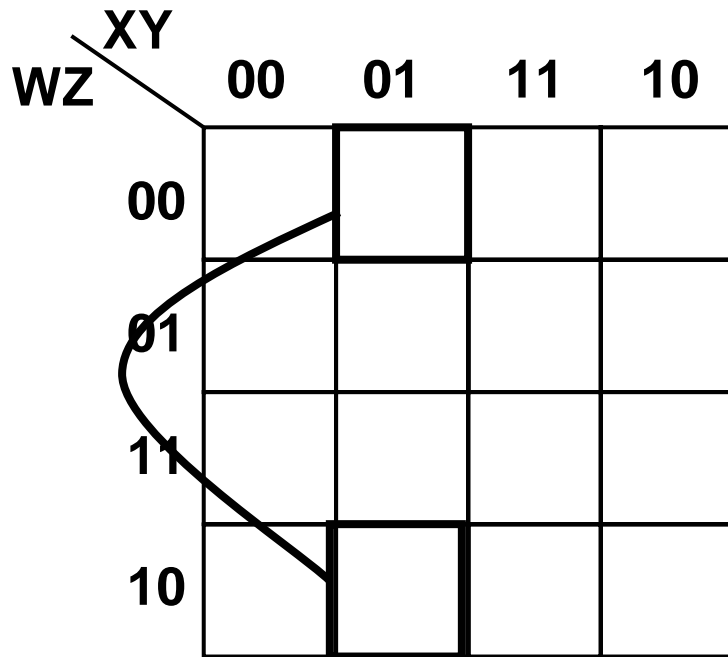
- Esempi celle adiacenti:

WZ \ XY	00	01	11	10
00				
01				
11				
10				

WZ \ XY	00	01	11	10
00				
01				
11				
10				

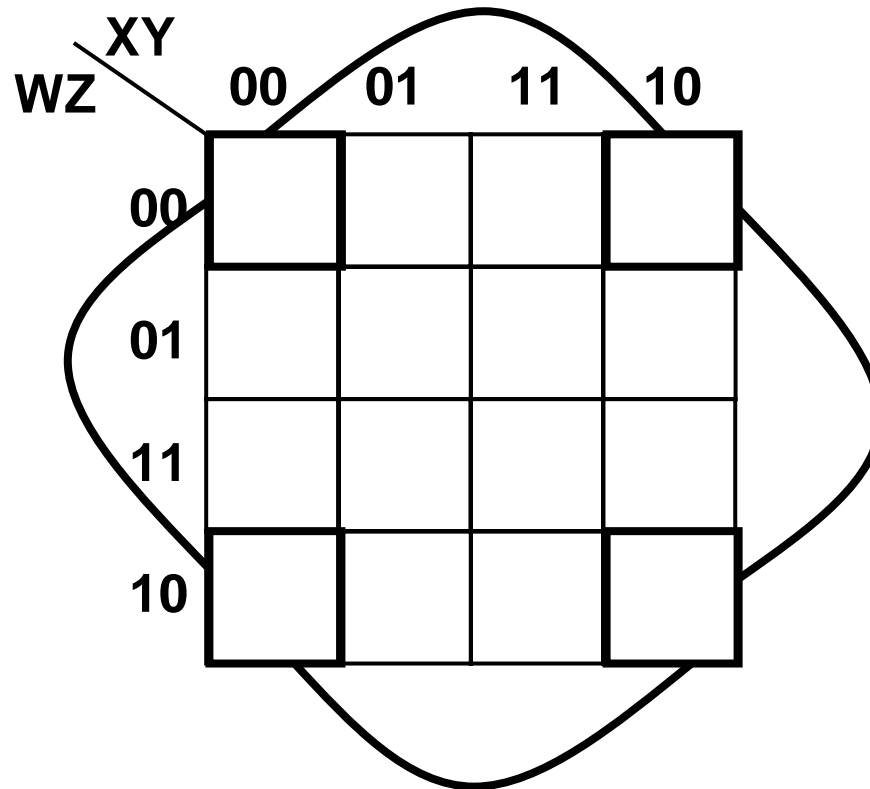
Regola per la copertura

- Esempi celle adiacenti:



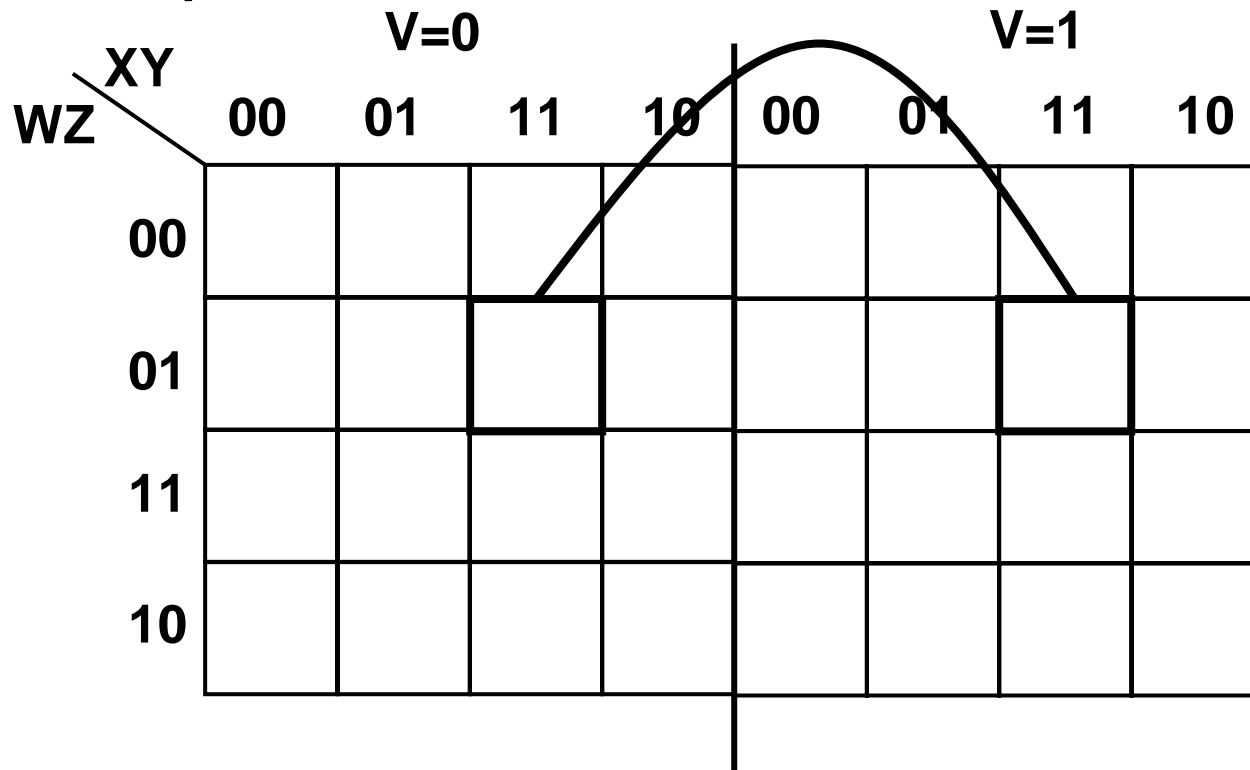
Regola per la copertura

- Esempi celle adiacenti:

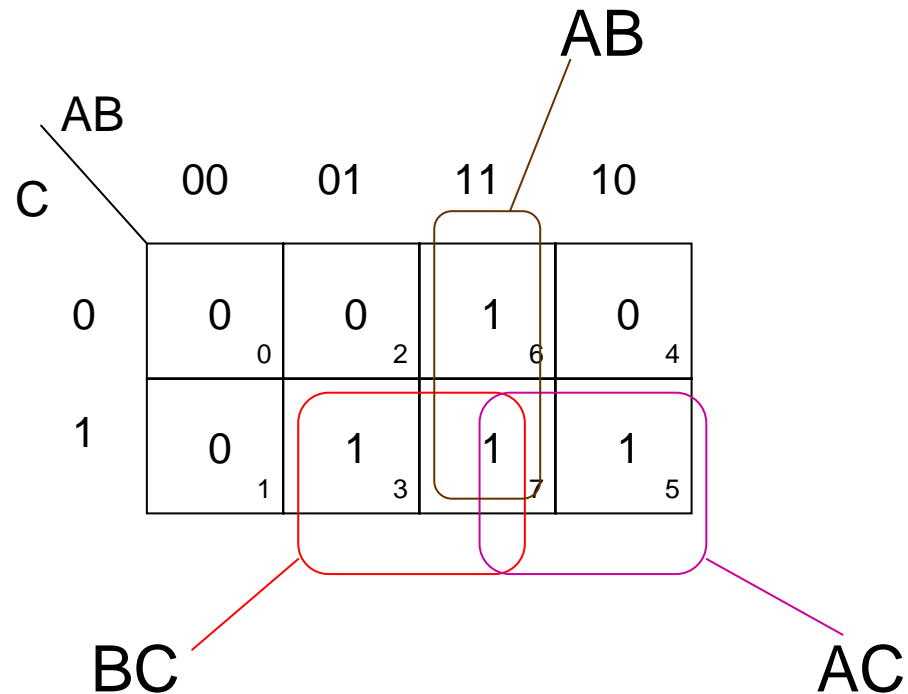


Regola per la copertura

- Esempi celle adiacenti:



Esempio di copertura



Mappe particolari

		A	
		0	1
B	0	0 0	1 2
	1	1 1	0 3

EXOR

		A,B			
		00	01	11	10
C	0	0 0	1 2	0 6	1 4
	1	1 1	0 3	1 7	0 5

		A,B			
		00	01	11	10
C,D	00	0 0	1 4	0 12	1 8
	01	1 1	0 5	1 13	0 9
	11	0 3	1 7	0 15	1 11
	10	1 2	0 6	1 14	0 10

Copertura SP

- Si coprono gli '1'
- Rimangono solo ingressi che non sono "variati" dentro uno stesso cubo
- Ingresso vale '1' \implies Compare in uscita il suo valore diretto (A)
- Ingresso vale '0' \implies Compare in uscita il suo valore negato (A')

Copertura PS

- Si coprono gli '0'
- Comportamento duale a copertura SP
 - Se ingresso '0' \implies ingresso preso "dritto"
 - Se ingresso '1' \implies ingresso è negato

Esempio di rete combinatoria

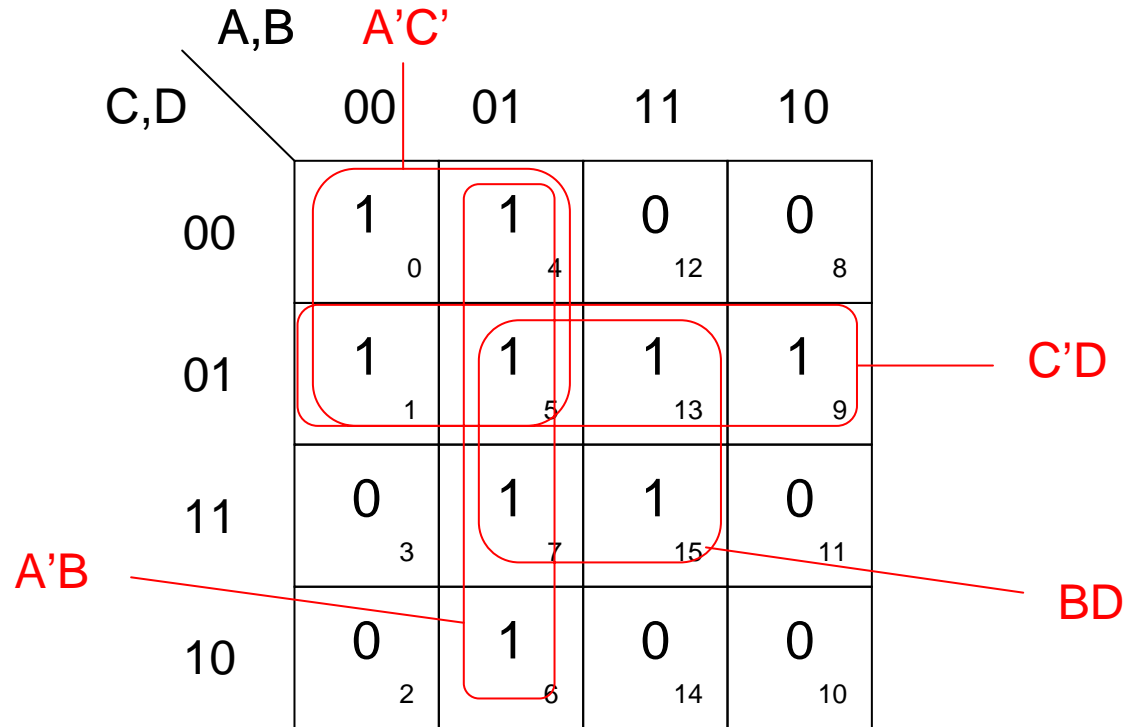
Tabella di verità

#	A	B	C	D	U
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	1

Mappa di Karnaugh

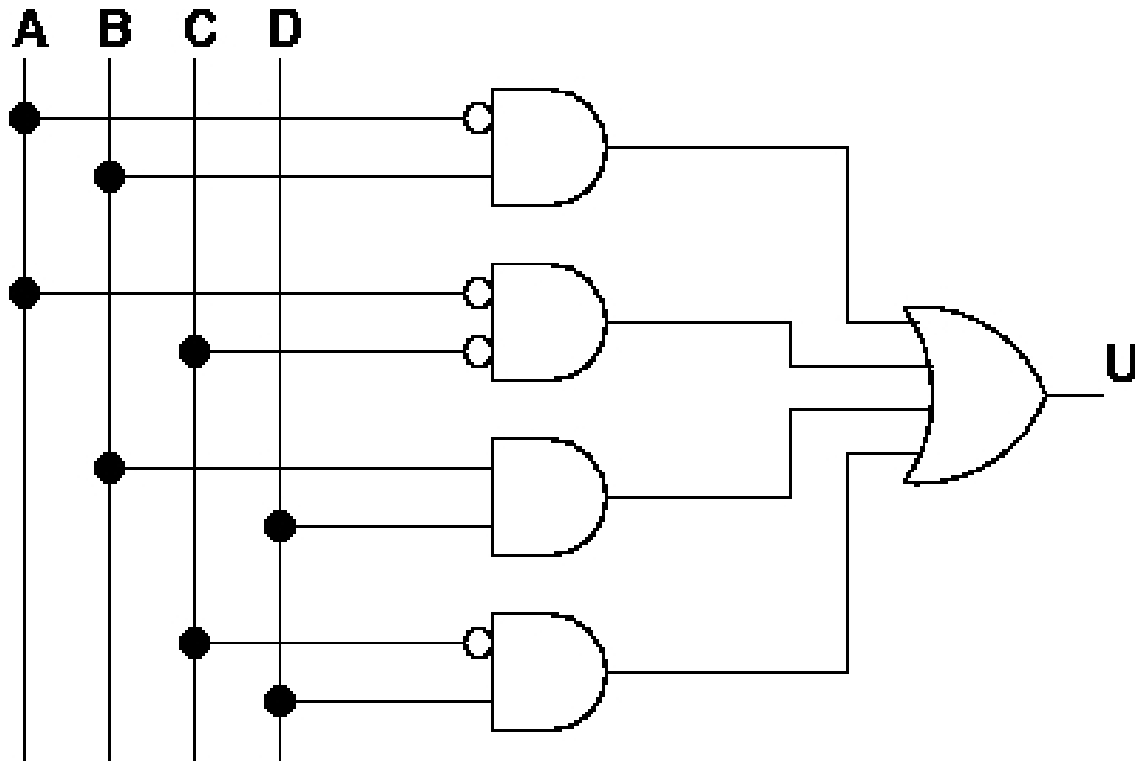
C,D		A,B			
		00	01	11	10
00	1 0	1 4	0 12	0 8	
01	1 1	1 5	1 13	1 9	
11	0 3	1 7	1 15	0 11	
10	0 2	1 6	0 14	0 10	

Copertura minima SP



$$U = \overline{A}B + \overline{A}\overline{C} + BD + \overline{C}D$$

Copertura minima SP



Copertura minima SP

C,D \ A,B		A,B			
		00	01	11	10
C,D	00	1 0	1 4	0 12	0 8
	01	1 1	1 5	1 13	1 9
	11	0 3	1 7	1 15	0 11
	10	0 2	1 6	0 14	0 10

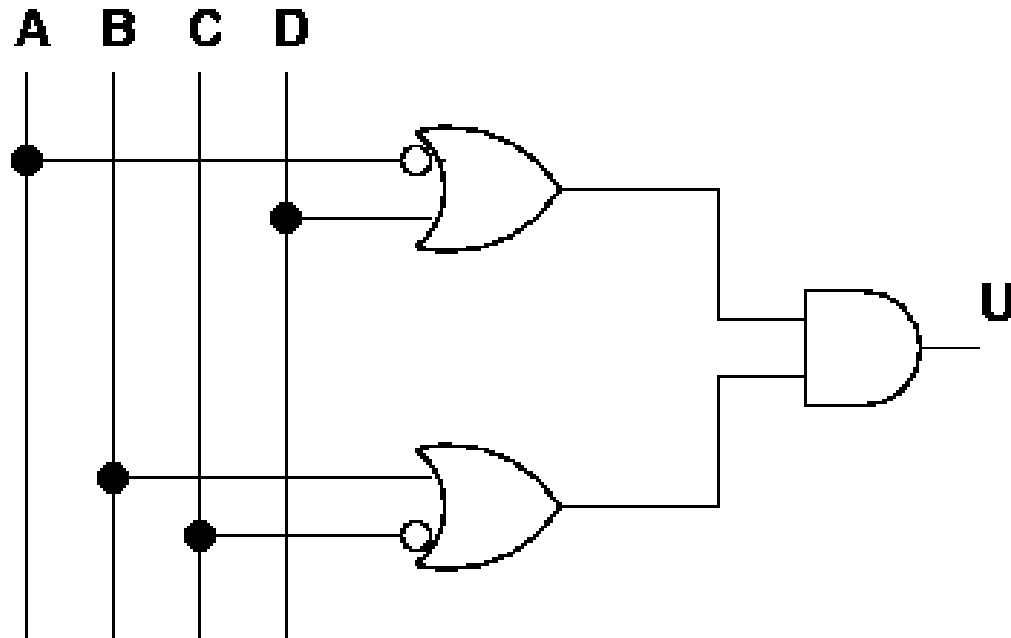
The Karnaugh map shows the function U with prime implicants circled in red. The prime implicants are:

- $B + C'$ (circles cells 0, 4, 1, 5)
- $A' + D$ (circles cells 12, 8, 15, 11, 14, 10)

 The minterm 7 is covered by both prime implicants.

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

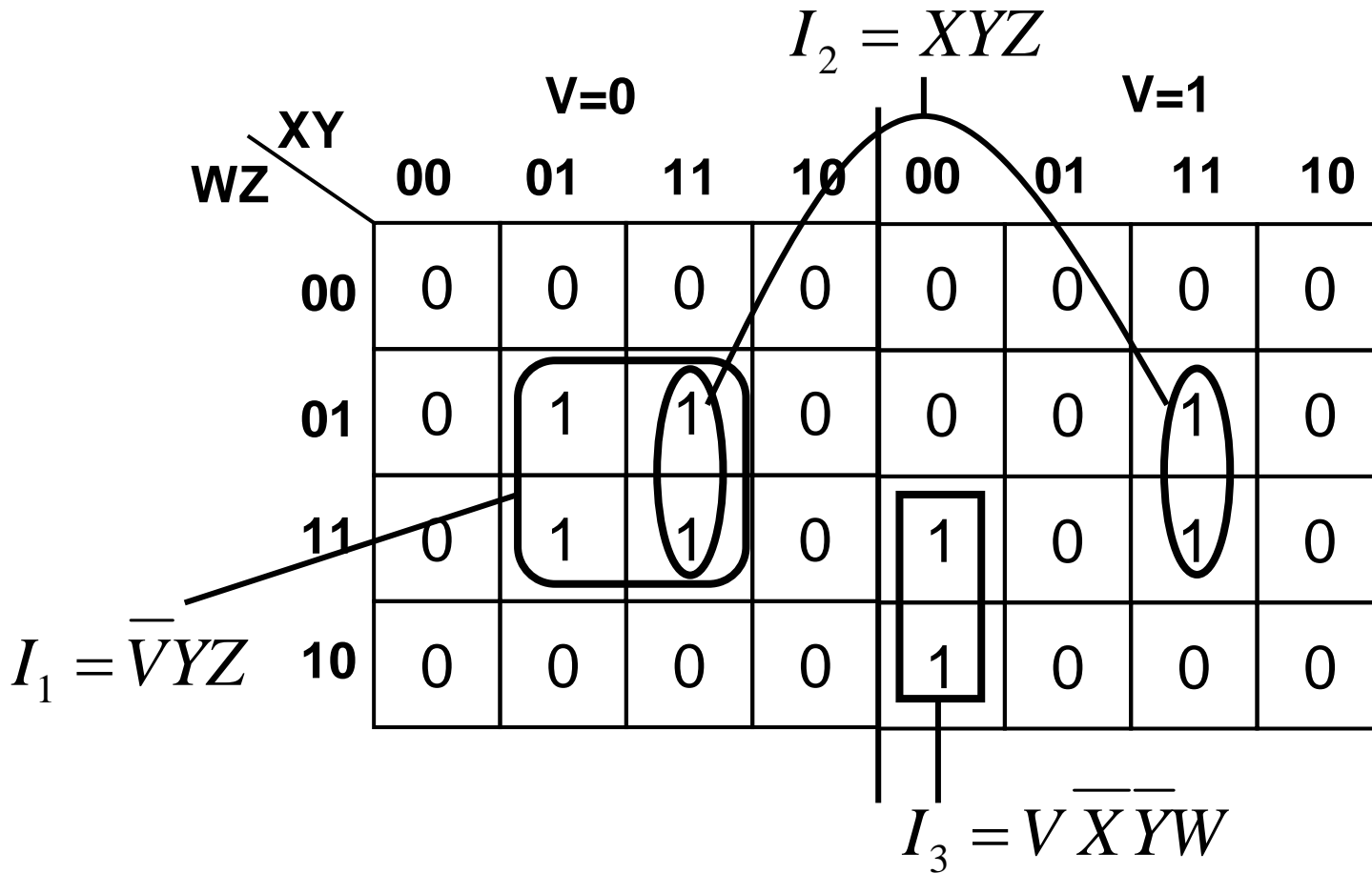
Copertura minima SP



Esempio

WZ \ XY		V=0				V=1			
		00	01	11	10	00	01	11	10
00	0	0	0	0	0	0	0	0	
01	0	1	1	0	0	0	1	0	
11	0	1	1	0	1	0	1	0	
10	0	0	0	0	1	0	0	0	

Esempio



$$U = \bar{V}YZ + XYZ + V \bar{X} \bar{Y} W$$

Sommario

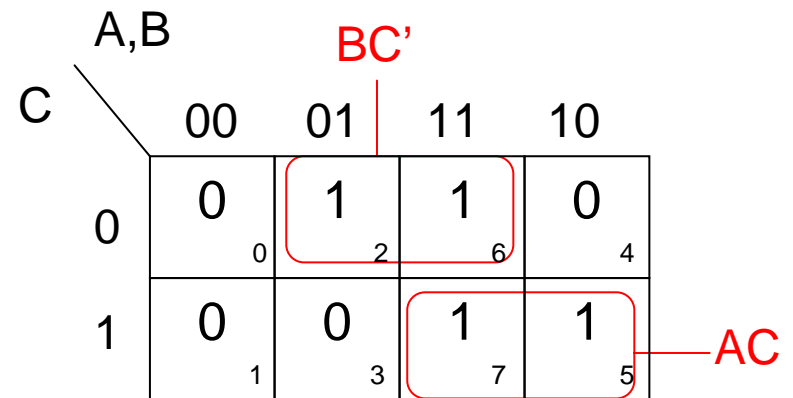
- Richiami di algebra booleana
- Mappe di Karnaugh
- Coperture SP & PS
- **Alee Statiche e loro minimizzazione**
- Circuiti sequenziali semplici
- Ulteriori circuiti

Esempio:

- Sia data la tabella di verità:

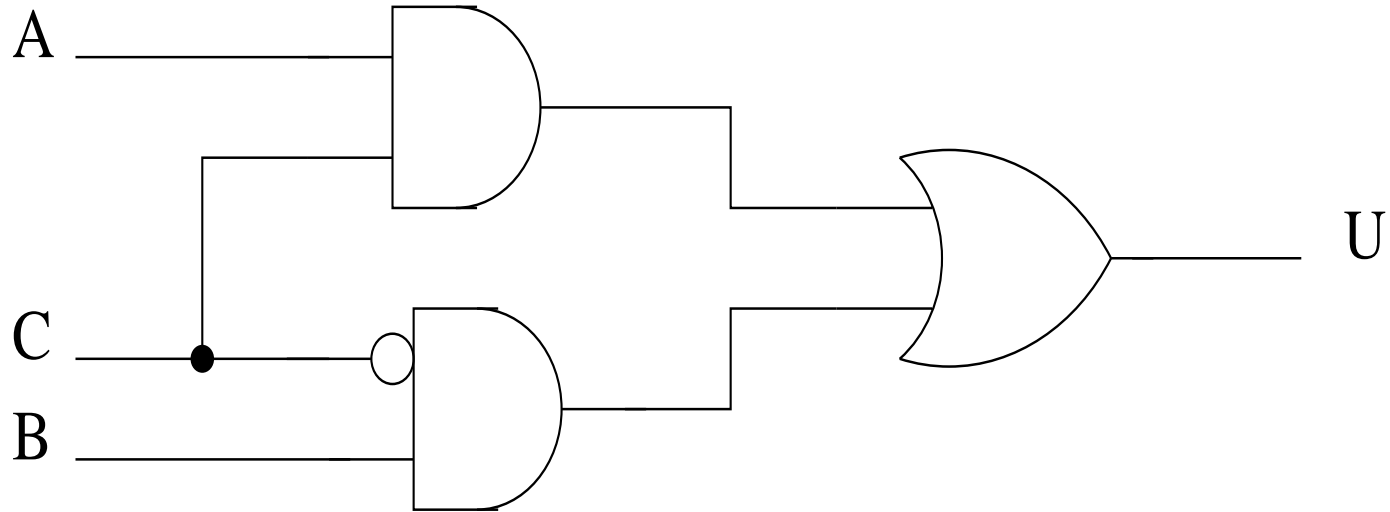
A	B	C	U
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	0	1	1

- Dalla mappa di Karnaugh si ricava

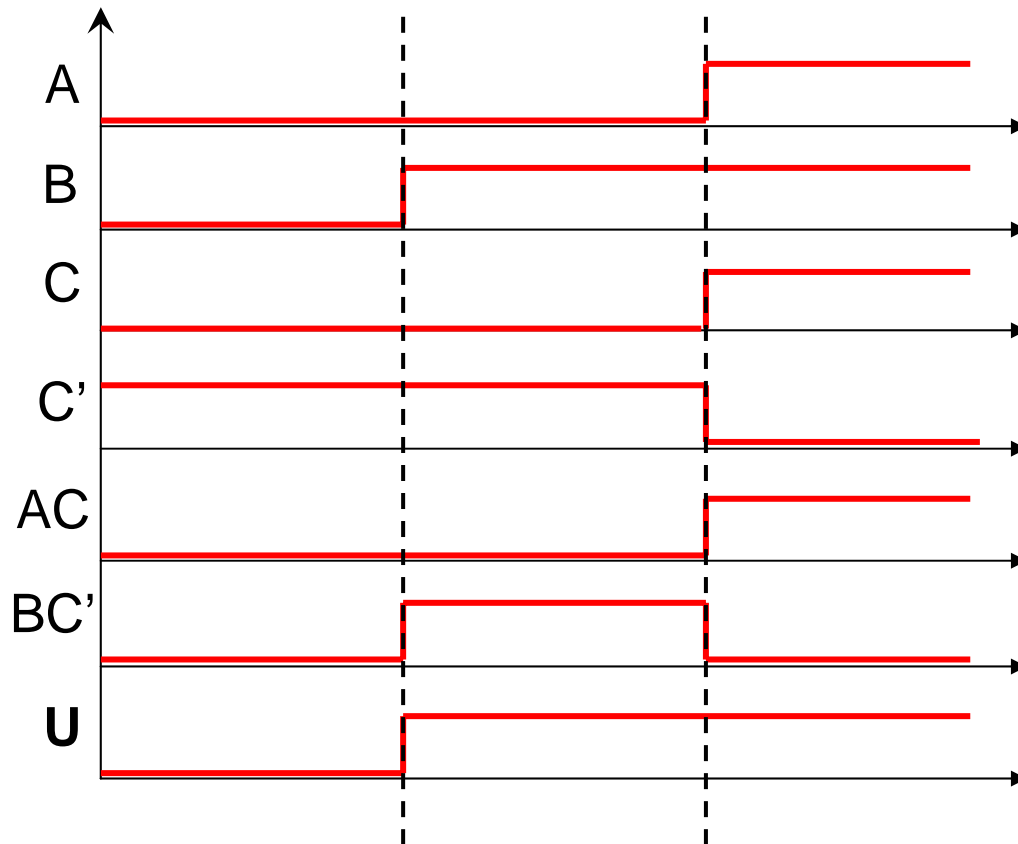


$$U = AC + BC'$$

Rete Combinatoria ottenuta

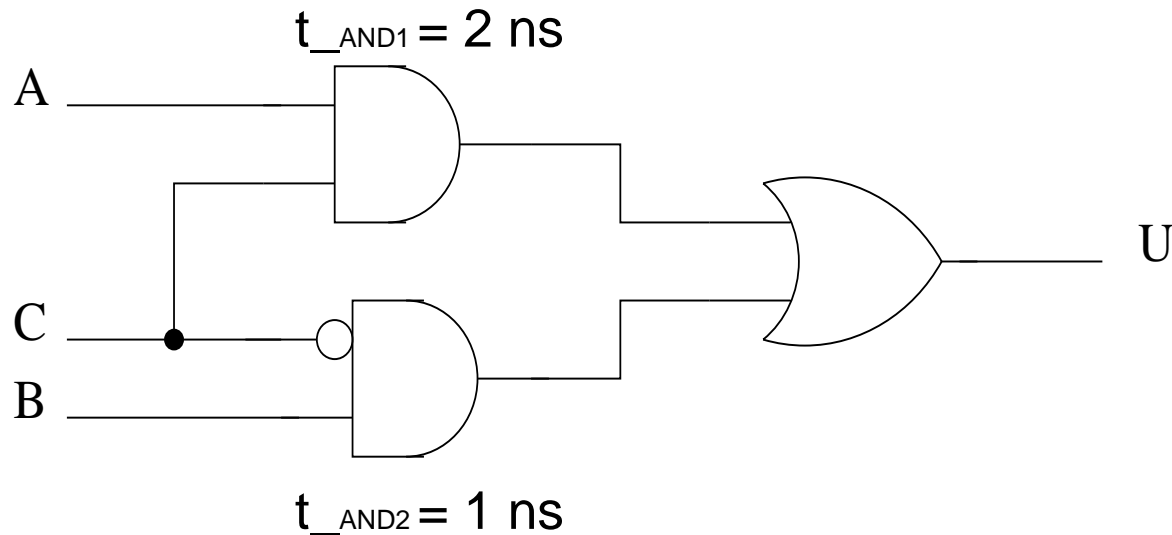


Comportamento ideale: porte logiche con ritardi nulli

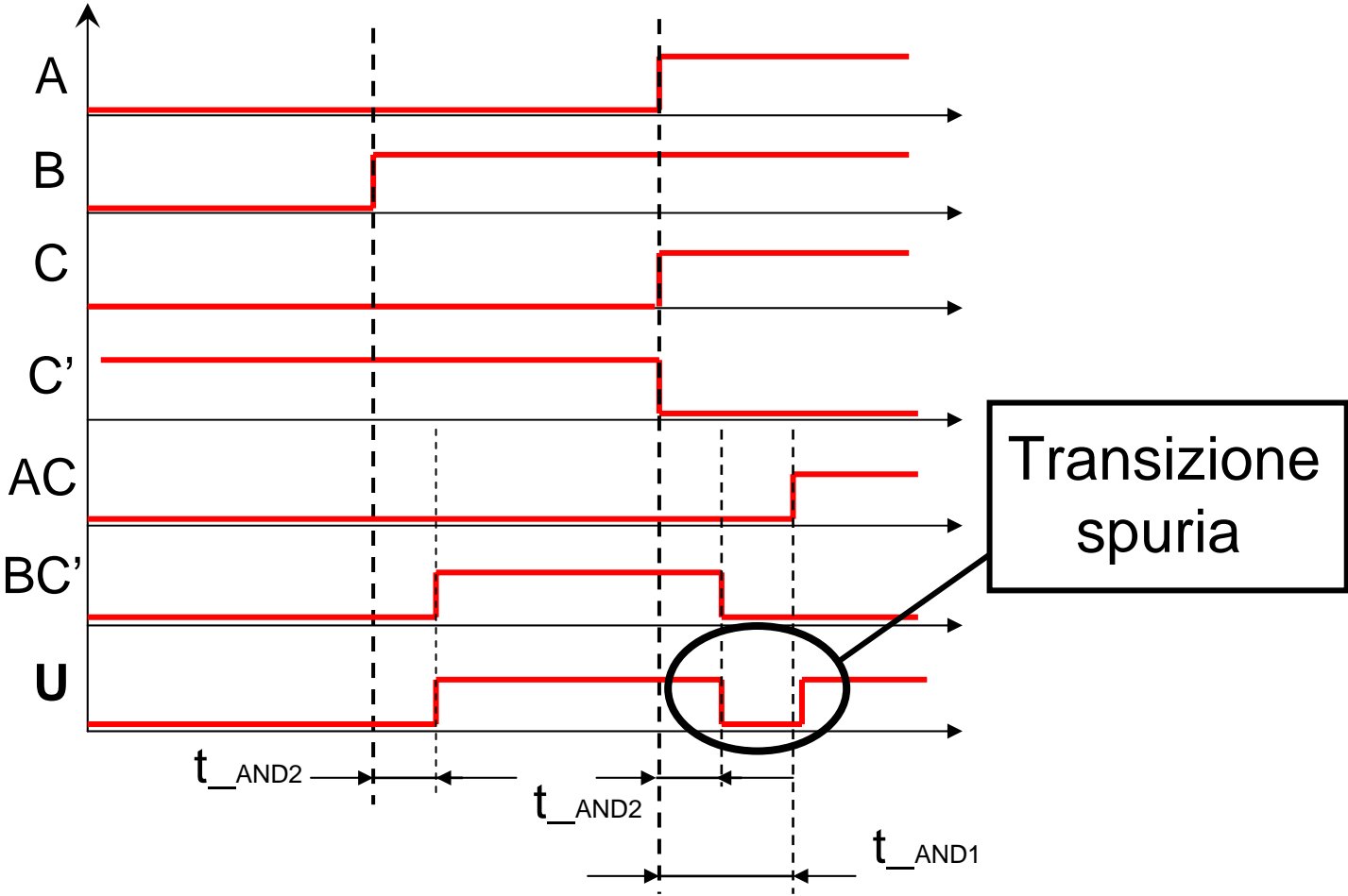


Porte con ritardi non nulli

- Supponiamo che le porte abbiano tempi di propagazione diversi.
- Ad esempio:



Comportamento del circuito

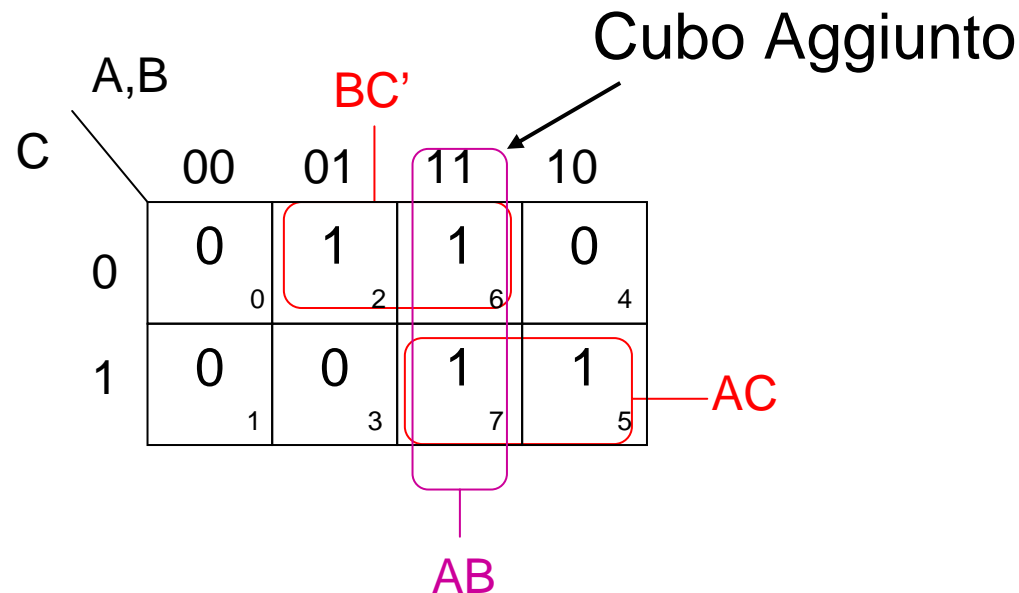


Analisi del comportamento

- I diversi tempi di ritardo causano transizioni spurie (“Glitch”)
- Sono dovuti al fatto che nella funzione logica non considero variazioni simultanee di + segnali
- Transizioni spurie causano anche maggiore dissipazione di potenza
- Per minimizzarne l’effetto si inserisce altra logica che mantiene fissi i segnali
- Non ho più funzioni minime

Copertura Hazard-Free

- Inserisco anche i “cubi” che coprono variazioni contemporanee di + segnali
- Esempio



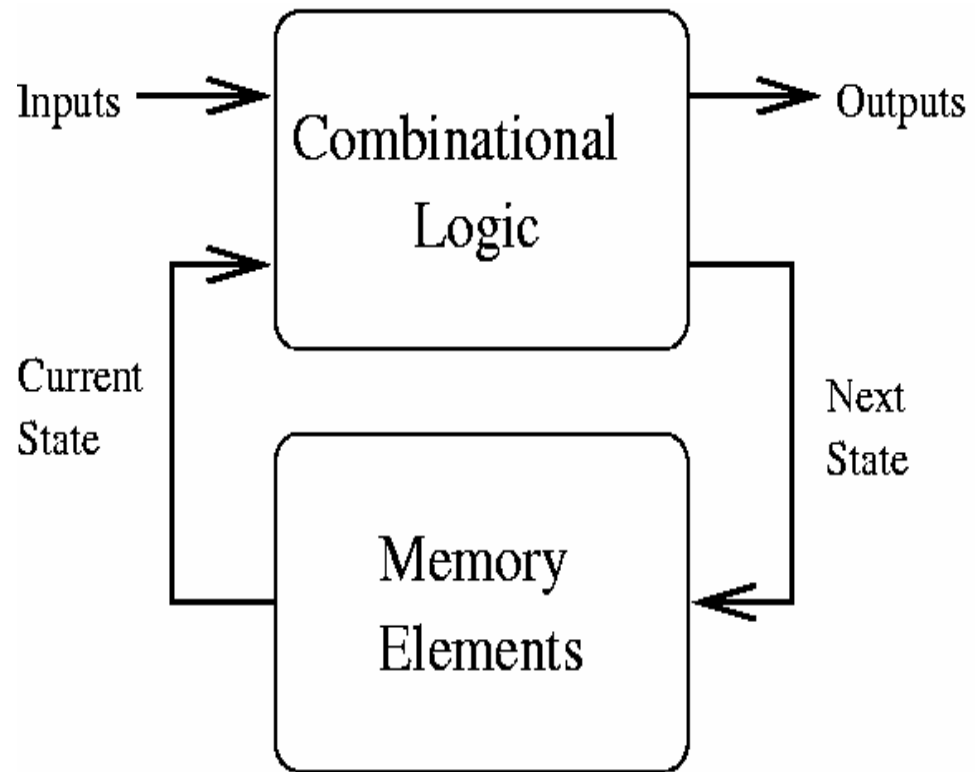
$$U = AB + AC + BC'$$

Sommario

- Richiami di algebra booleana
- Mappe di Karnaugh
- Coperture SP & PS
- Alee Statiche e loro minimizzazione
- **Circuiti sequenziali semplici**
- Ulteriori circuiti

Concetti preliminari

- Circuiti digitali sono costituiti da una parte di “elaborazione” (Rete combinatori) ed una parte di controllo e temporizzazione (“Parte sequenziale”)

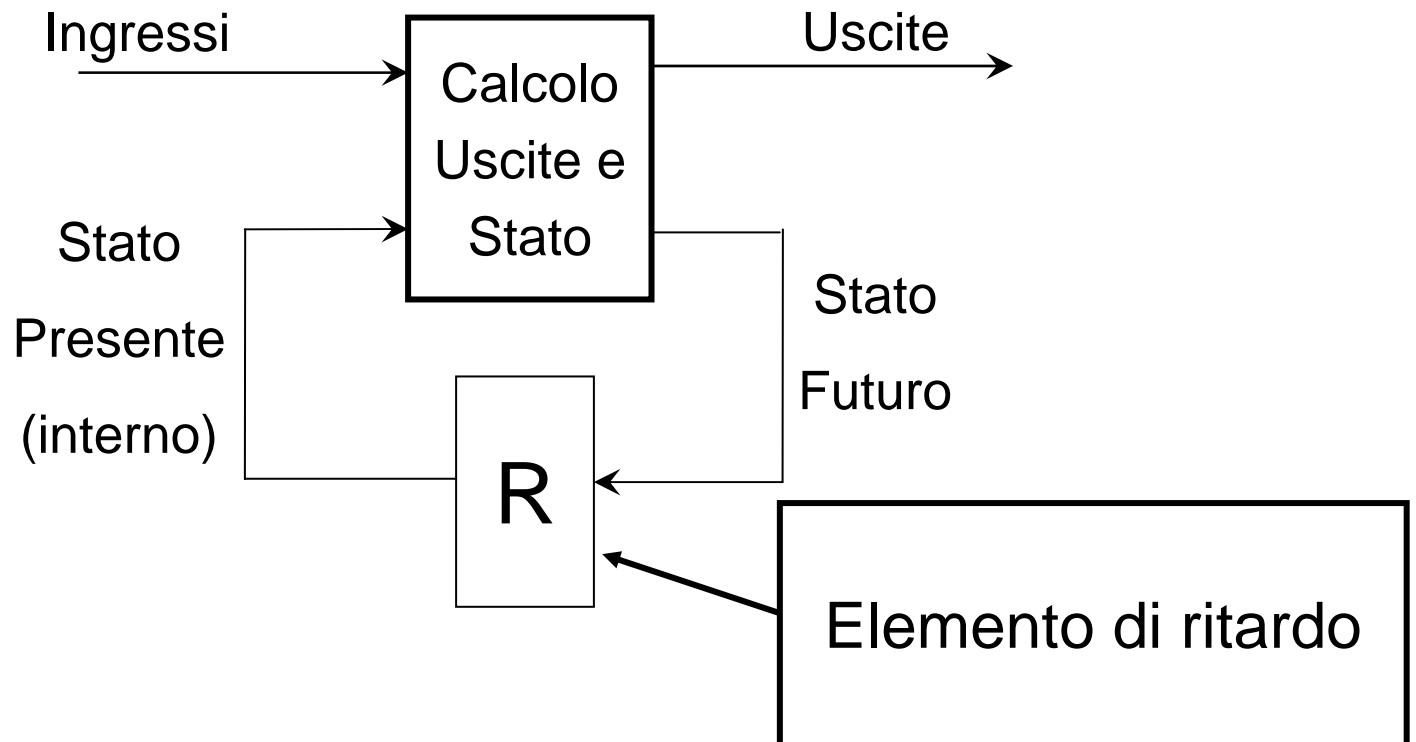


Concetti Preliminari

- Le uscite dei circuiti sequenziali dipendono: dagli ingressi attuali, ma anche dagli INGRESSI PASSATI
- Sono cioè circuiti con MEMORIA
- Il circuito si trova in uno STATO PRESENTE
- Evolverà in uno STATO FUTURO

Concetti Preliminari

- Un circuito sequenziale generico sarà costituito da (modello di Huffman)

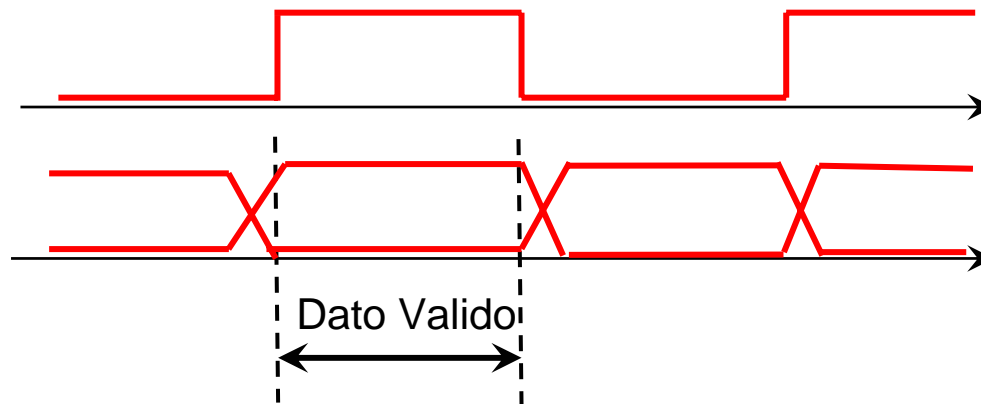


Concetti preliminari

- Tipi principali di circuiti sequenziali:
- **Circuiti Asincroni:** lo stato può cambiare in ogni istante (in base a ritardi intrinseci porte)
- **Circuiti Sincroni:** lo stato può cambiare solo in corrispondenza di un segnale di cadenza

Concetti preliminari

- La maggioranza dei circuiti integrati digitali progettati sono “CIRCUITI SINCRONI”
- Esiste un segnale di cadenza (chiamato spesso “clock”) che “valida” e sincronizza i segnali considerati
- Esempio:

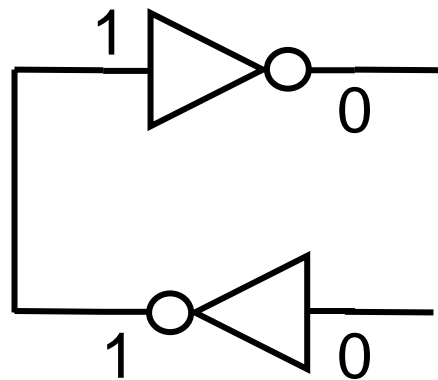


Concetti preliminari

- I segnali devono mantenersi “stabili” a seguito di un “fronte” del segnale di cadenza
- Si devono inserire dei circuiti logici che memorizzino lo stato dei segnali

Concetti preliminari

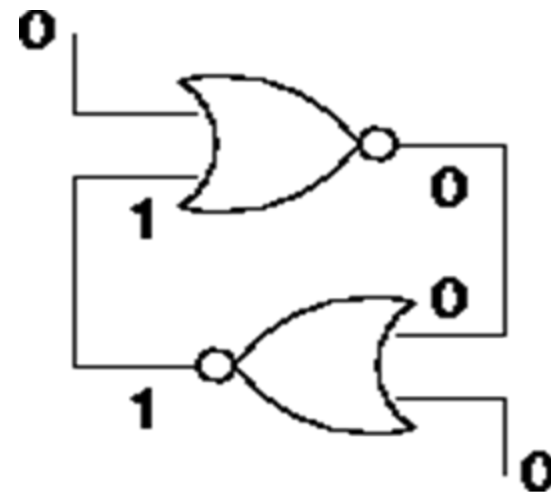
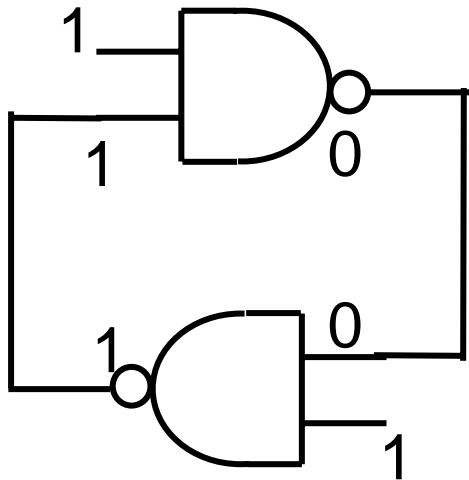
- L'elemento di memorizzazione/ritardo si può ricondurre ad un anello di inverter



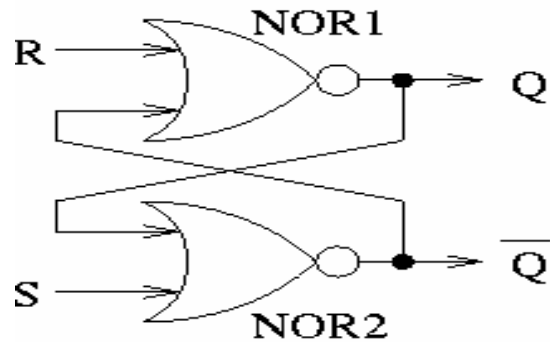
Il dato memorizzato non si può modificare!!!

Concetti preliminari

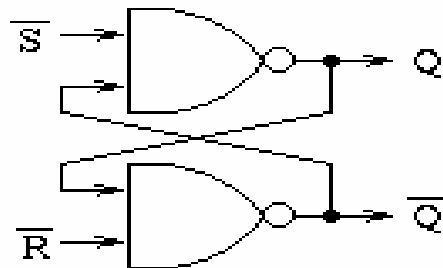
- Al fine di “inserire” valori dall’esterno le NOT si possono sostituire con porte che possono “simulare” lo stesso comportamento:
 - NAND oppure NOR



Latch SR

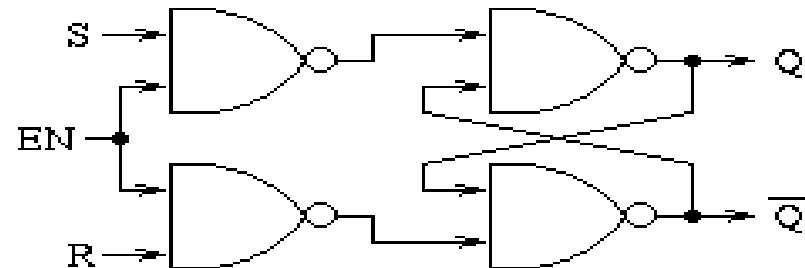


R	S	$Q(t+1)$	$\overline{Q(t+1)}$
0	0	$Q(t)$	$\overline{Q(t)}$
0	1	1	0
1	0	0	1
1	1	0	0 (disallowed)



\overline{S}	\overline{R}	$Q(t+1)$	$\overline{Q(t+1)}$
0	0	1	1 (disallowed)
0	1	1	0
1	0	0	1
1	1	$Q(t)$	$\overline{Q(t)}$

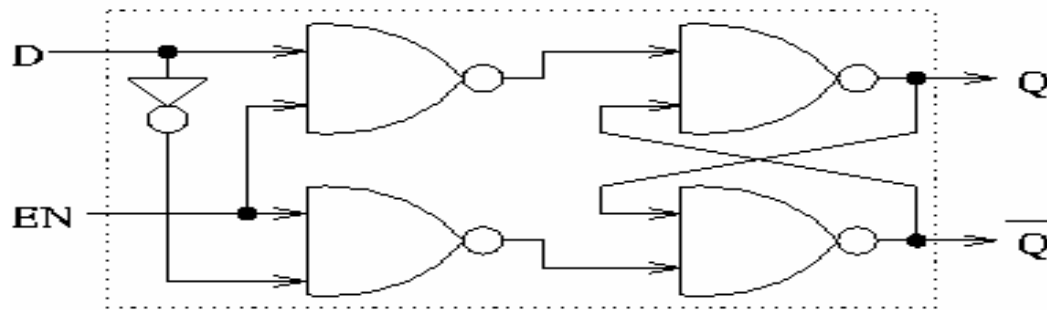
(a)



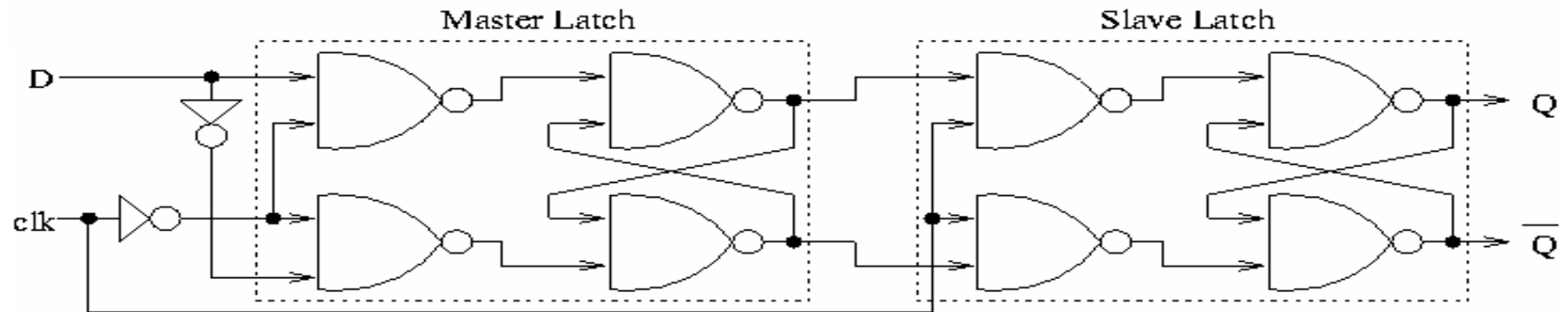
EN	S	R	$Q(t+1)$	$\overline{Q(t+1)}$
0	x	x	$Q(t)$	$\overline{Q(t)}$
1	0	0	$Q(t)$	$\overline{Q(t)}$
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1 (disallowed)

(b)

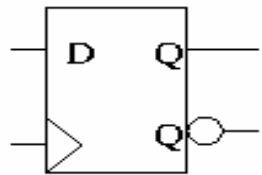
Latch e flip-flop D



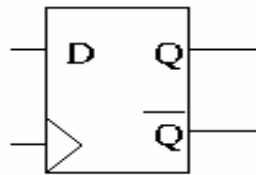
EN	D	$Q(t+1)$	$\overline{Q(t+1)}$
0	x	$Q(t)$	$\overline{Q(t)}$
1	0	0	1
1	1	1	0



(a)



(b)

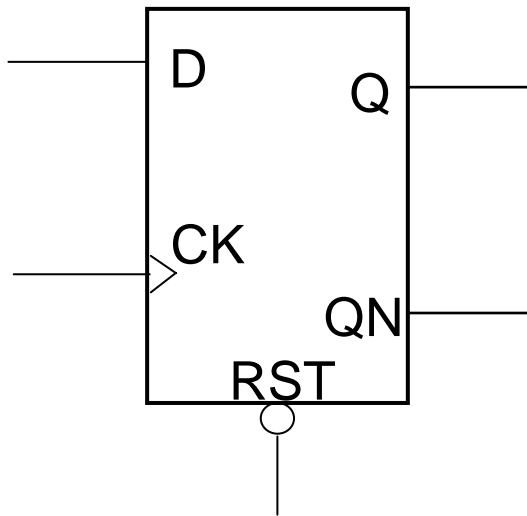


(c)

clk	D	$Q(t+1)$	$\overline{Q(t+1)}$
0	x	$Q(t)$	$\overline{Q(t)}$
1	x	$Q(t)$	$\overline{Q(t)}$
1	0	0	1
1	1	1	0

(d)

FLIP-FLOP D

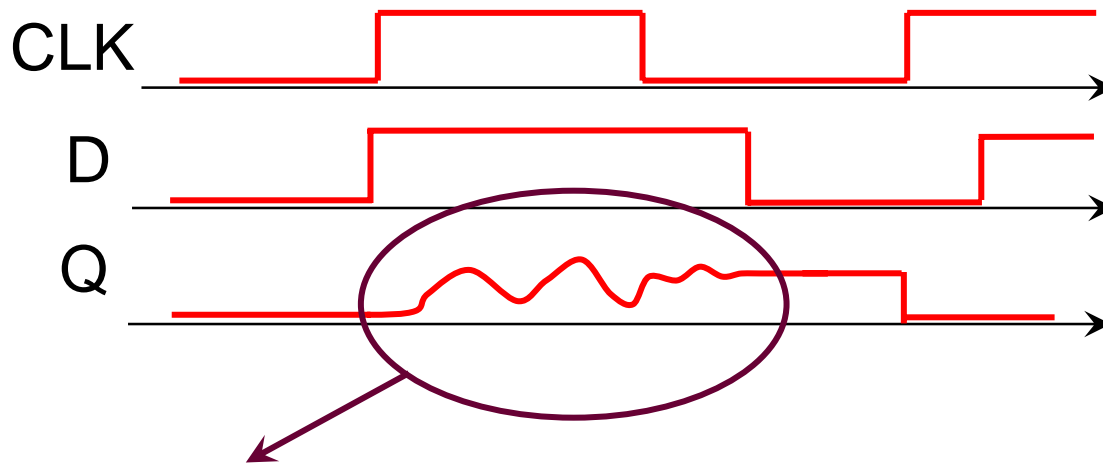


- Tabella di verità:

RST	CK	D	Q _n	QN _n
0	-	-	0	1
1		0	0	1
1		1	1	0

Tempistica del Flip-Flop

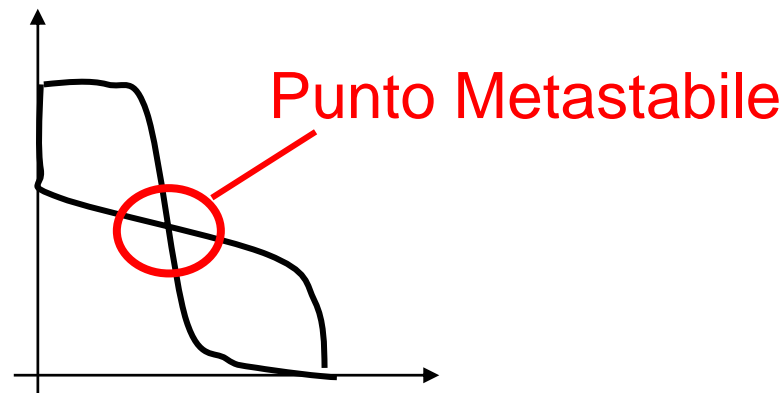
- Il dato D è “campionato” sul “fronte di salita” del clock
- Se D varia durante tale fronte cosa succede?



Oscillazione dell'uscita di durata non predicibile a priori \Rightarrow **METASTABILITA'**

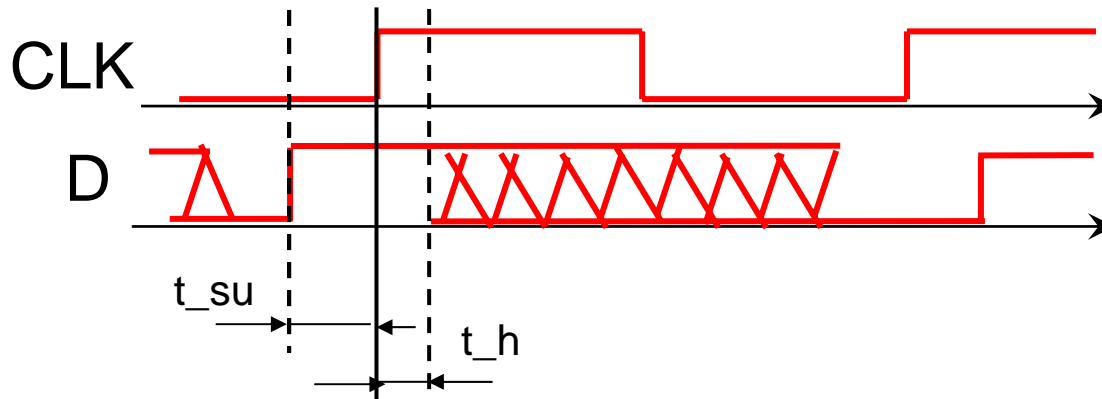
Tempistica del Flip-Flop

- Metastabilità è dovuta alla presenza di un punto quasi-stabile (“metastabile”) nella transcaratteristica ingresso-uscita dell’anello di inverter
- Se campione ingresso con valore vicino a quel punto impiego del tempo per “risolvere” metastabilità
- Più campione vicino a punto metastabile, più lungo sarà il tempo con uscita non definita

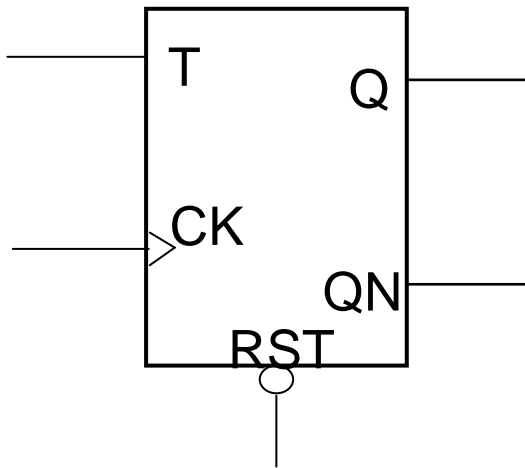


Tempi di set-up e hold

- Per evitare metastabilità \implies dato deve essere “stabile” prima e dopo fronte del clock
- Tempo min. per cui dato deve essere stabile prima del clock \implies tempo di set-up (t_{su})
- Tempo min. per cui dato deve essere stabile dopo il clock \implies tempo di hold (t_h)



Flip-Flop T

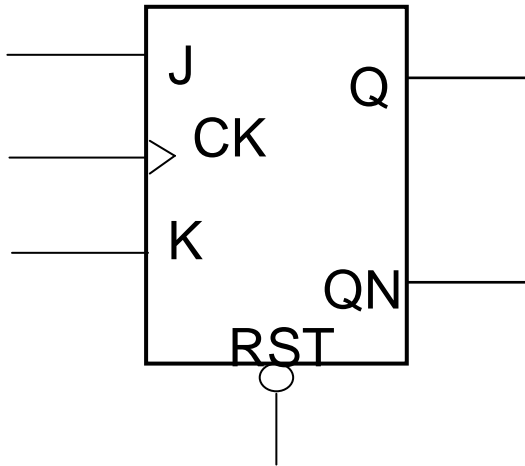


- Tabella di verità:

RST	CK	T	Q _n	QN _n
0	-	-	0	1
1		0	Q _{n-1}	QN _{n-1}
1		1	QN _{n-1}	Q _{n-1}

- Se ingresso T='1' \Rightarrow l'uscita Q varia, alternandolo, il suo valore ad ogni colpo di CLK ("toggle")
- Si dimezza la frequenza di clock

Flip-Flop J-K

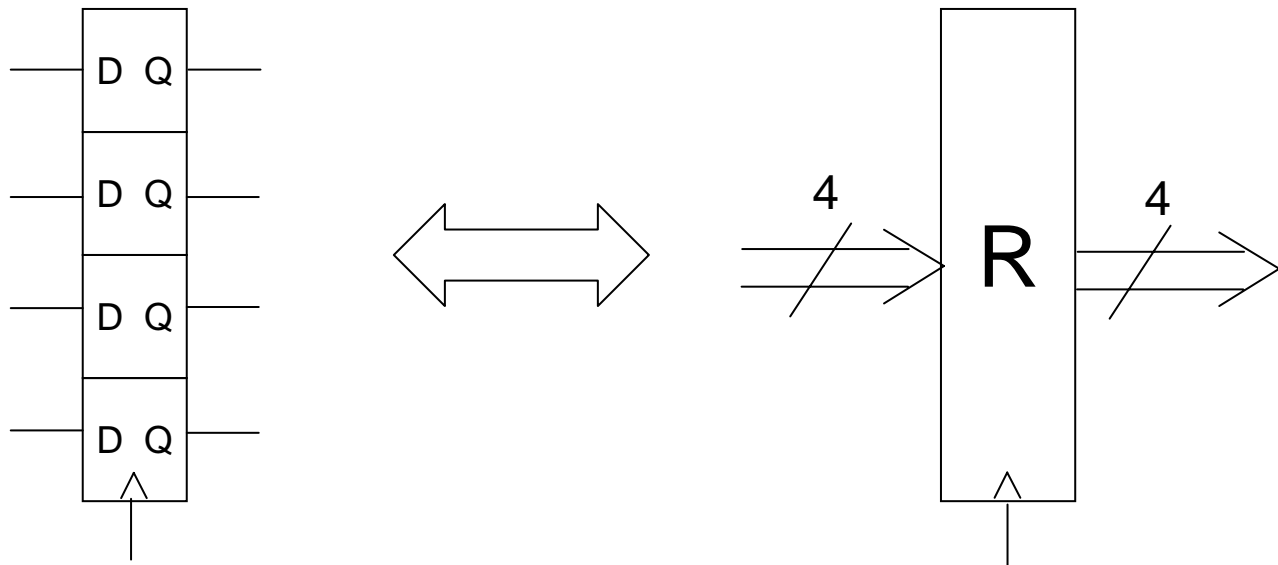


- Tabella di verità:

RST	CK	J	K	Q_n	QN_n
0	-	-	-	0	1
1	\uparrow	0	0	Q_{n-1}	QN_{n-1}
1	\uparrow	0	1	0	1
1	\uparrow	1	0	1	0
1	\uparrow	1	1	QN_{n-1}	Q_{n-1}

Registri

- Affiancando n flip-flop D si ottiene un registro
- Esempio Registro a 4 bit:

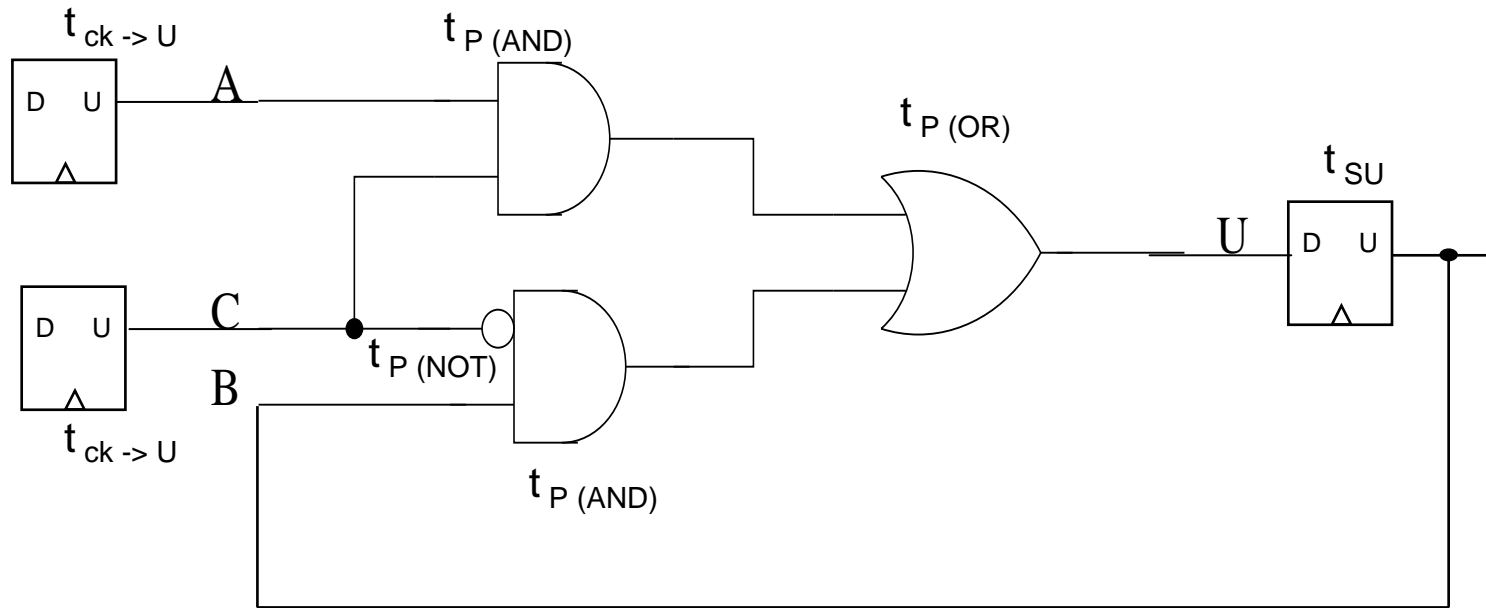


Calcolo Frequenza Massima

- Identifico i percorsi combinatori che collegano elementi sequenziali (cammini tra FF)
- Sommo i ritardi di propagazione delle porte combinatorie e del FF da cui esce un segnale
- Sommo il tempo di set-up del FF a cui segnale arriva
- Se ho percorsi paralleli che si ricongiungono considero il tempo massimo

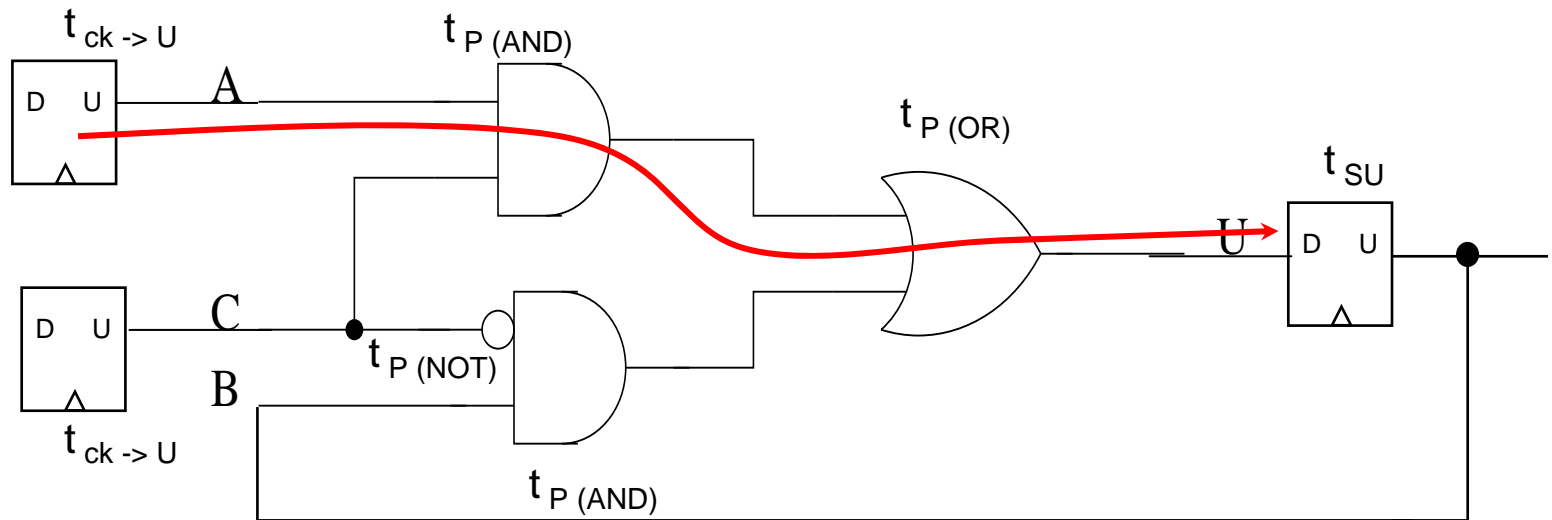
Calcolo Frequenza Massima

- Esempio



Calcolo Frequenza Massima

- Percorso 1:

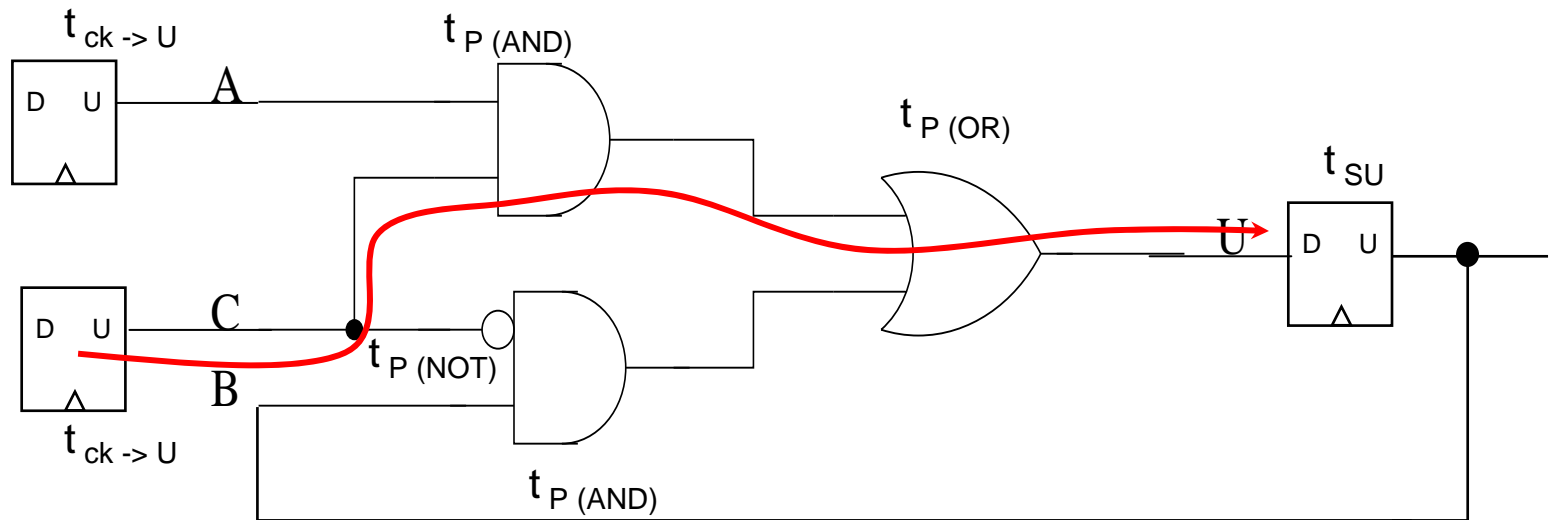


$$T_1 = t_{ck \rightarrow U} + t_{P(AND)} + t_{P(OR)} + t_{SU}$$

$$T_1 = 5 \text{ ns} + 2 \text{ ns} + 2 \text{ ns} + 4 \text{ ns} = 13 \text{ ns}$$

Calcolo Frequenza Massima

- Percorso 2:

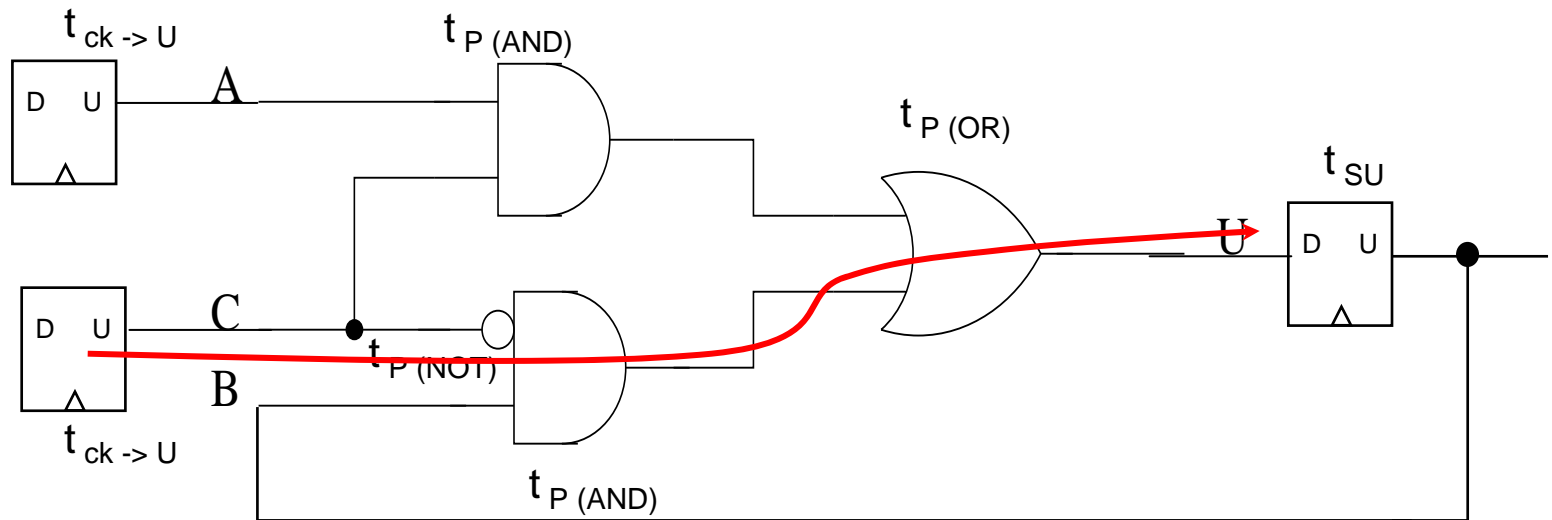


$$T_2 = t_{ck \rightarrow U} + t_{P(AND)} + t_{P(OR)} + t_{SU}$$

$$T_2 = 5 \text{ ns} + 2 \text{ ns} + 2 \text{ ns} + 4 \text{ ns} = 13 \text{ ns}$$

Calcolo Frequenza Massima

- Percorso 3:

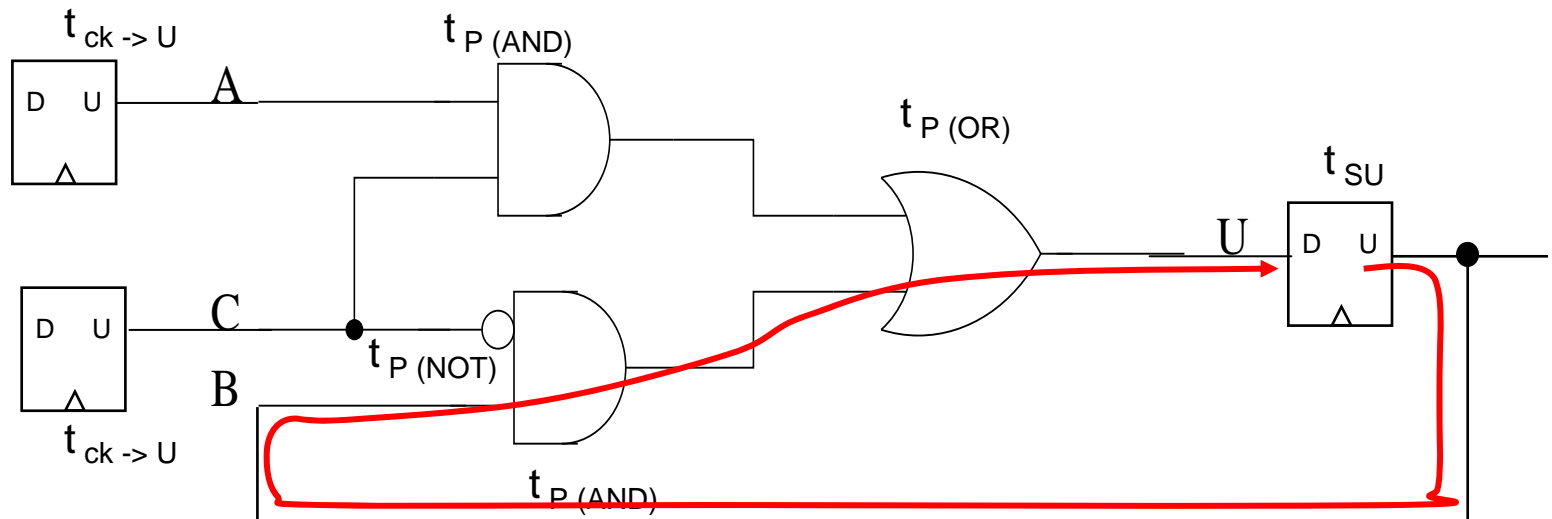


$$T_3 = t_{ck \rightarrow U} + t_{P(NOT)} + t_{P(AND)} + t_{P(OR)} + t_{SU}$$

$$T_3 = 5 \text{ ns} + 0.5 \text{ ns} + 2 \text{ ns} + 2 \text{ ns} + 4 \text{ ns} = 13.5 \text{ ns}$$

Calcolo Frequenza Massima

- Percorso 4:

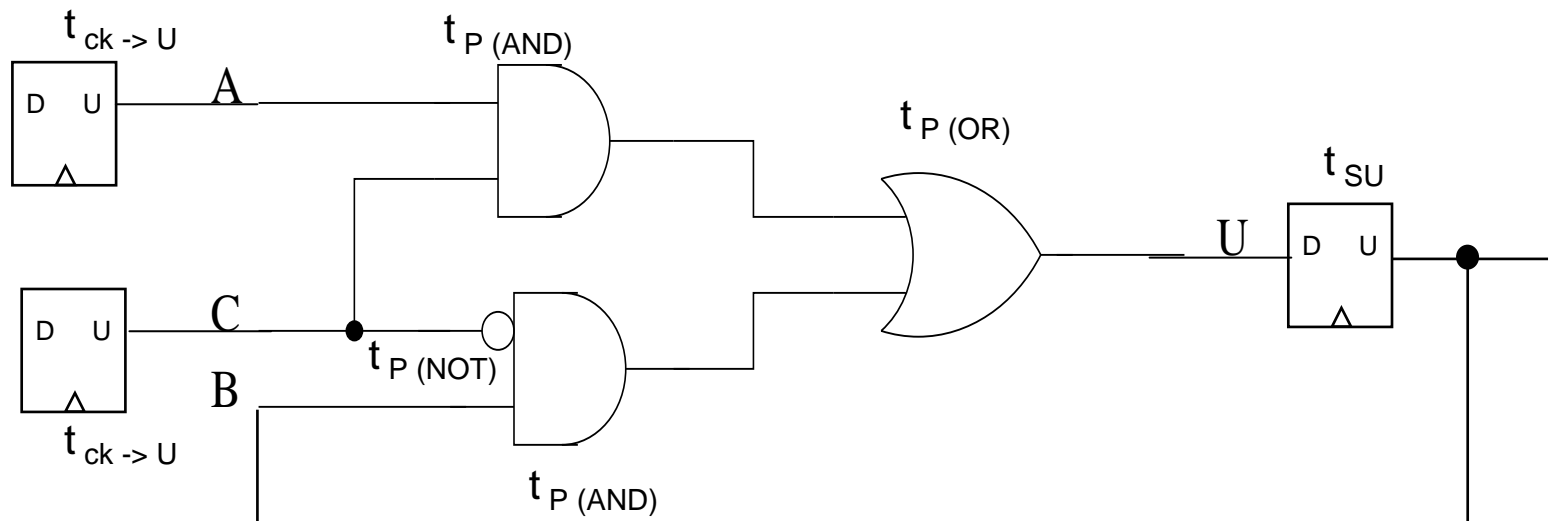


$$T_4 = t_{ck \rightarrow U} + t_{P(AND)} + t_{P(OR)} + t_{SU}$$

$$T_4 = 5 \text{ ns} + 2 \text{ ns} + 2 \text{ ns} + 4 \text{ ns} = 13.5 \text{ ns}$$

Calcolo Frequenza Massima

- Periodo Minimo di clock richiesto:



$$T_{\min} = \max(T_1, T_2, T_3, T_4) = T_3 = 13.5 \text{ ns}$$

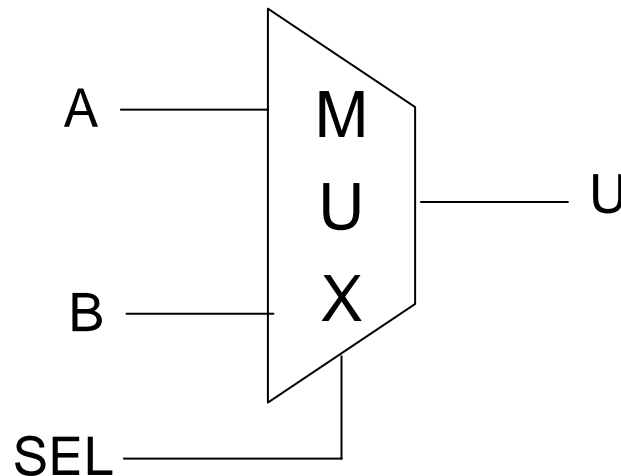
$$f_{\max} = \frac{1}{T_{\min}} = 74,07 \text{ MHz}$$

Sommario

- Richiami di algebra booleana
- Mappe di Karnaugh
- Coperture SP & PS
- Alee Statiche e loro minimizzazione
- Circuiti sequenziali semplici
- **Ulteriori circuiti**

Multiplexer

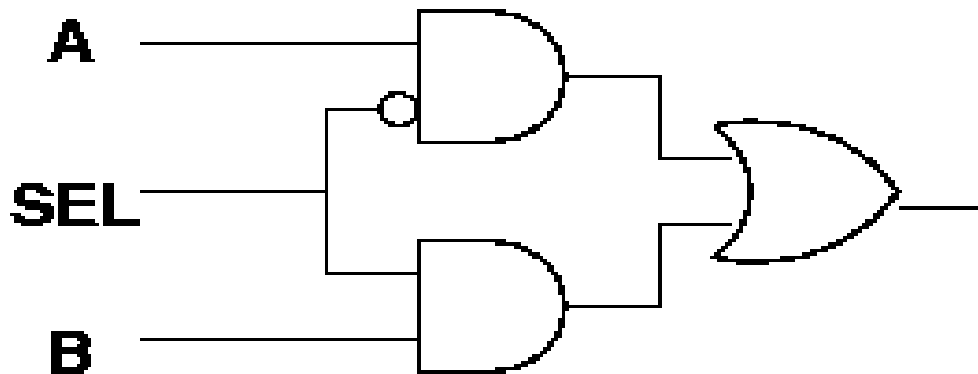
- Seleziona tra n ingressi
- Necessita di $\log_2(n)$ segnali di selezione
- Esempio MUX a 2 ingressi:



Multiplexer

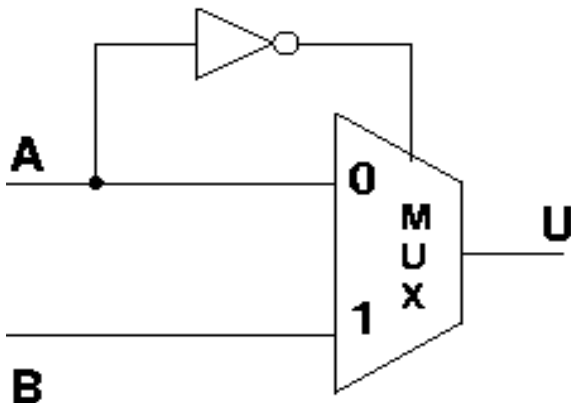
- Funzione logica

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



Multiplexer

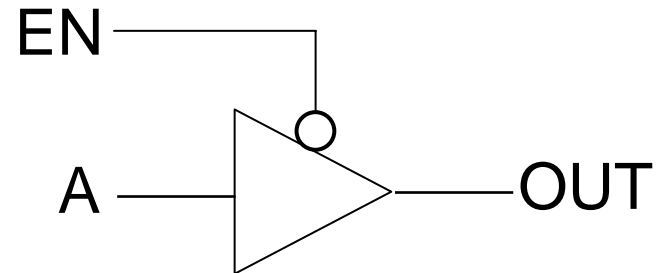
- Con i MUX e porte NOT si possono realizzare qualsiasi funzione logica
- Es:



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

Driver Tri-State

- Si utilizzano per pilotare bus bidirezionali
- Presentano un ingresso di abilitazione “enable” (EN)
- Se è attivo uscita corrisponde ad ingressi
- Se è disattivato uscita è in Alta Impedenza (HiZ)

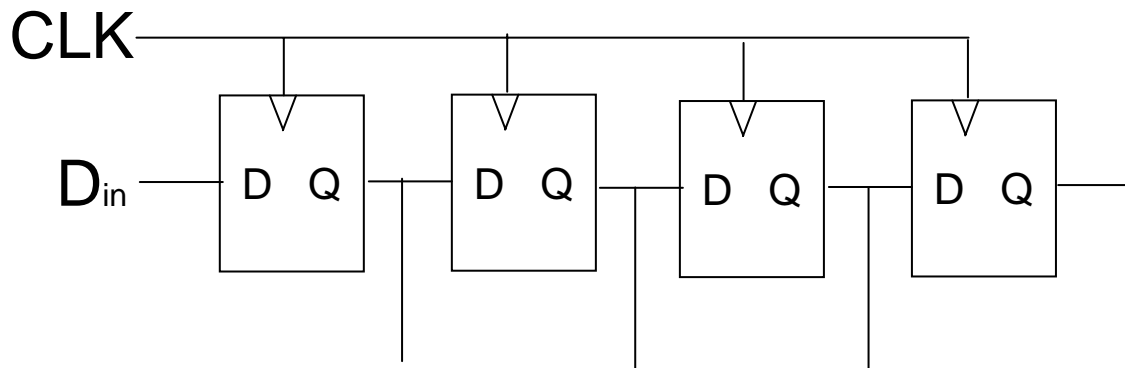


- Tabella di verità

EN	A	OUT
0	0	0
0	1	1
1	-	HiZ

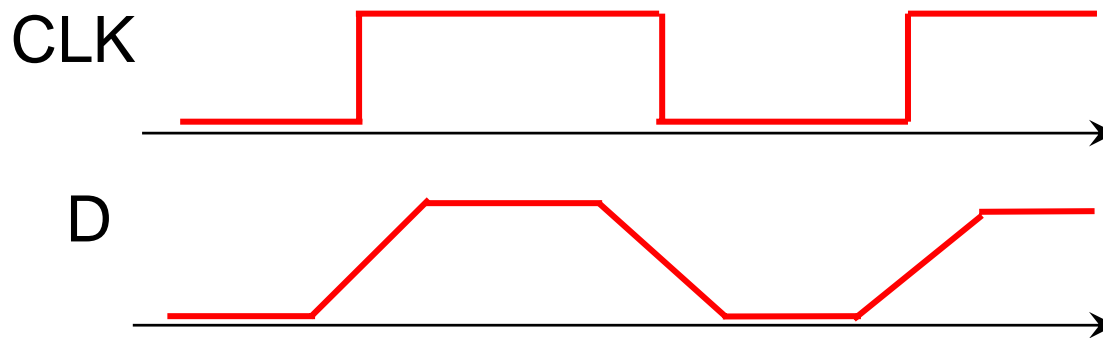
Shift Register

- Serie di flip-flop D in cui l'uscita di 1 corrisponde all'ingresso del successivo
- Esempio Shift Register a 4 bit



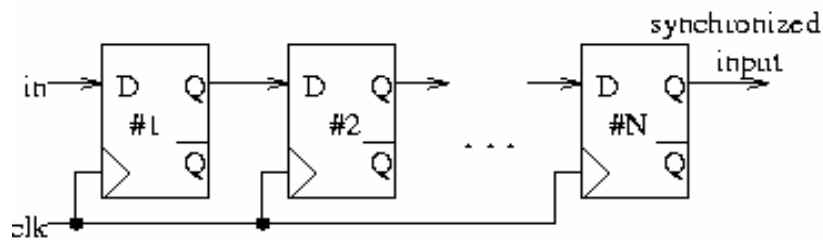
Sincronizzatore

- Metastabilità non si può eliminare completamente, ma si può cercare di ridurre
- Tale problema è molto più evidente per gli ingressi provenienti dall'esterno che variano + lentamente

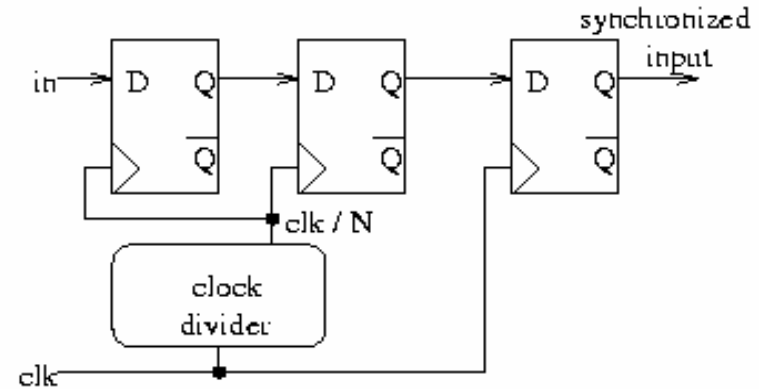


Sincronizzatore

- Si possono far passare gli ingressi in una catena di flip-flop
- Si riduce notevolmente la probabilità di entrare in metastabilità



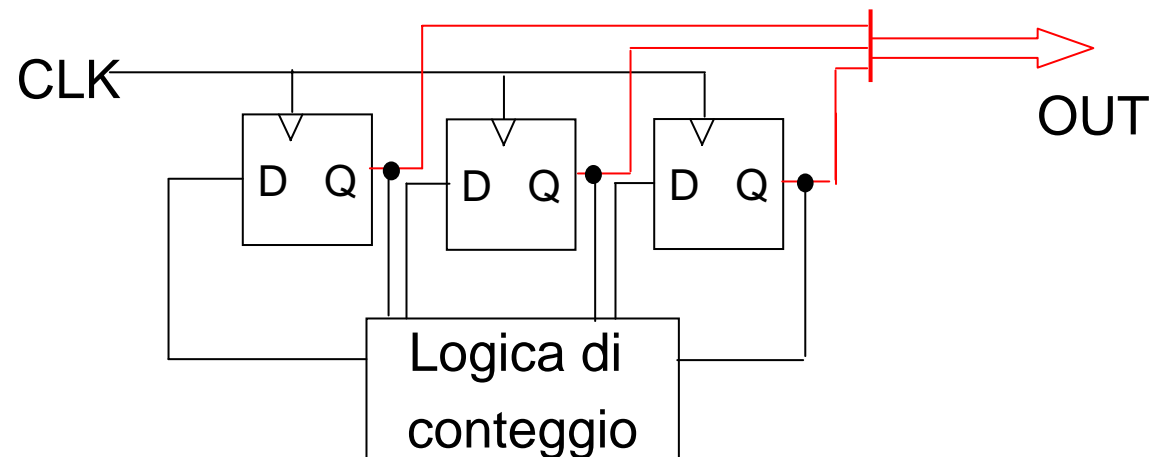
(a)



(b)

Contatore

- “Conta” un numero 2^n di colpi di clock (n è il numero di flip-flop utilizzati)
- Esempio Contatore UP modulo 8 (3 flip-flop)

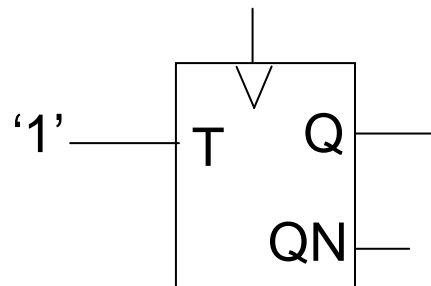


Esempio progetto logica di conteggio

- Analizziamo andamento delle uscite:

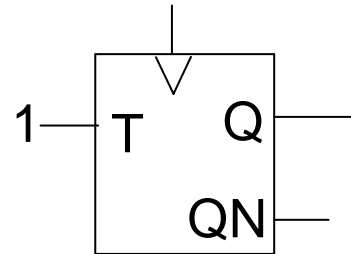
000 → 001 → 010 → 011 → ...

- L'ultimo bit varia ad ogni colpo di clock
- Comportamento tipo flip-flop T ("toggle")
- Lo realizziamo:



Esempio:

- Usiamo flip-flop T:



- Uscita Q varia quando ingresso è '1'
- Da tabella di verità $T1=Q0$
- Da tabella di verità: $T2=Q0*Q1$
- Fine conteggio (TC): $TC=Q0*Q1*Q2$

Esempio:

