

# RETI LOGICHE T

Ingegneria Informatica

## Esercitazione 1

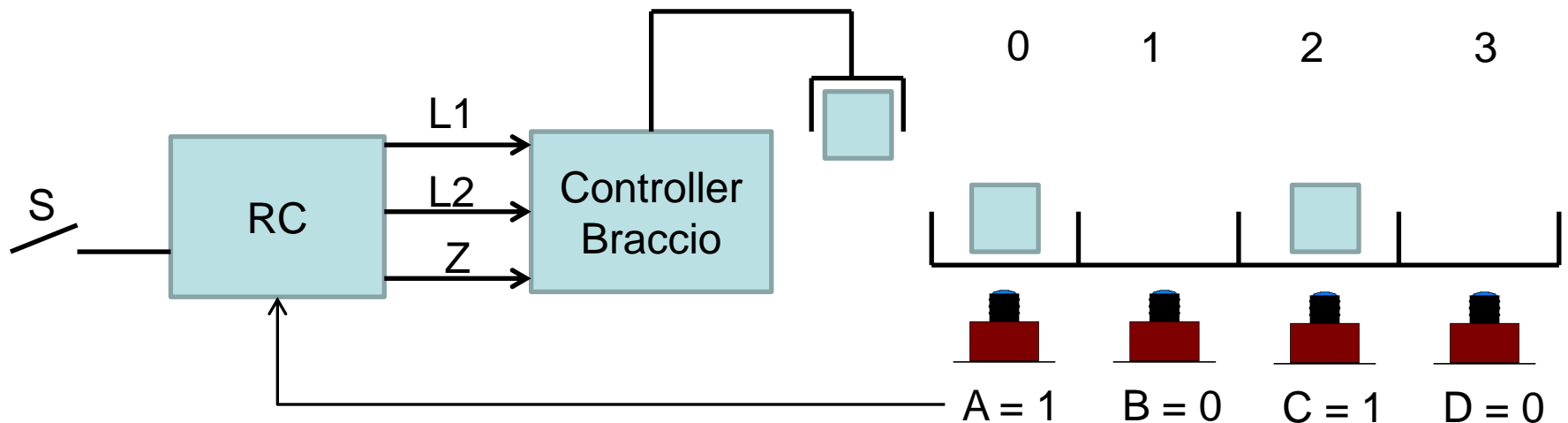
### Reti Combinatorie

Marco Lippi ([marco.lippi3@unibo.it](mailto:marco.lippi3@unibo.it))

[Lucidi realizzati da Samuele Salti]

# Esercizio Sintesi RC

- ◆ Una macchina di pallettizzazione automatica ha bisogno di conoscere in ogni momento la prima posizione libera sulle 4 disponibili in cui caricare il nuovo pezzo prodotto.
- ◆ In ogni posizione è montata una fotocellula che indica se la posizione è libera (0) o occupata (1). Indichiamo con (A,B,C,D) le fotocellule da sx a dx.
- ◆ L'operatore può decidere tramite un selettore S se il riempimento deve avvenire nella prima posizione libera a destra (0) o a sinistra (1).
- ◆ Il braccio robotico necessita in input di una coppia di segnali L1, L2 che indichino la posizione libera in cui depositare il prossimo pezzo e di un segnale Z=1 se non ci sono posti liberi. Se non vi sono posti liberi, il braccio ignora gli input L1, L2.



# Esercizio Sintesi RC

- ◆ Individuare:
  - ◆ La tabella della verità delle funzioni richieste
  - ◆ La sintesi con un MUX a 4 vie dei segnali L1, L2
  - ◆ La sintesi tramite rete di costo minimo SP
  - ◆ La sintesi della rete di costo minimo con soli gate NAND
- ◆ Verificare se le reti minime individuate soffrono del fenomeno dell'alea statica, e nel caso indicare le modifiche necessarie ad evitarlo
- ◆ Sintetizzare con una PAL assegnata le reti di costo minimo prive di alea statica

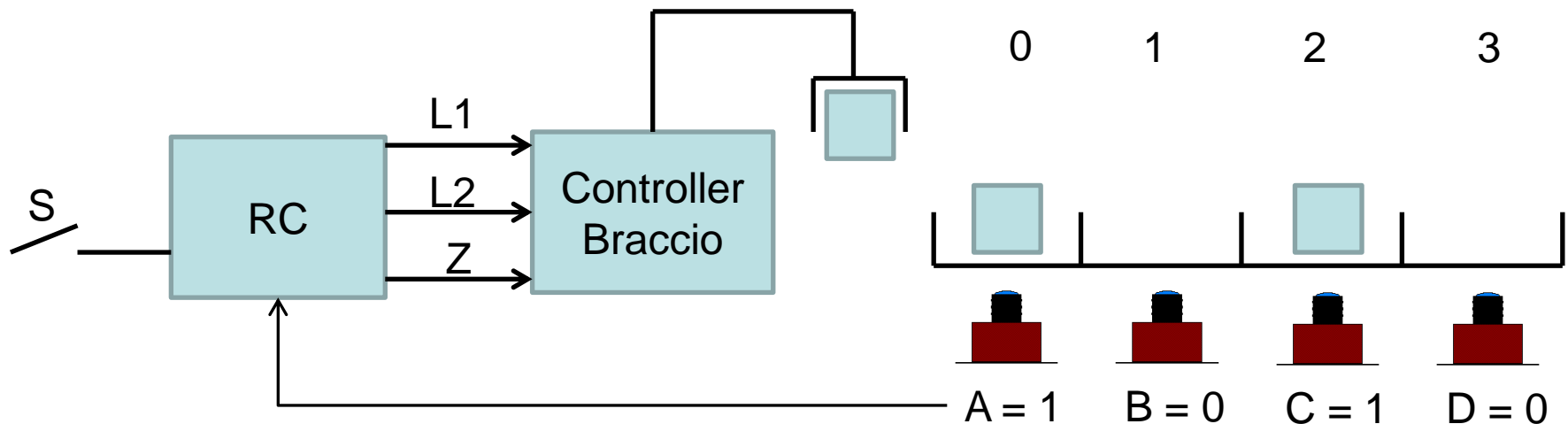
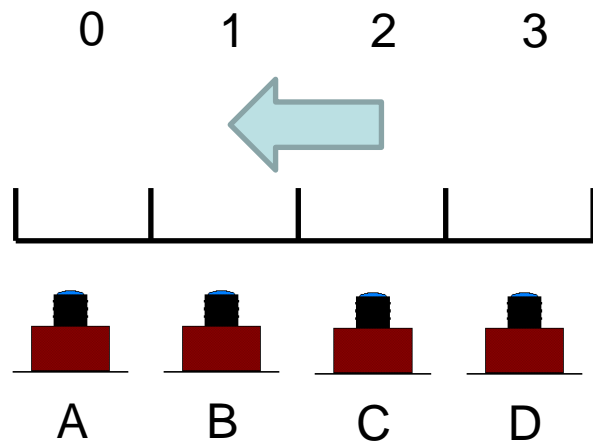
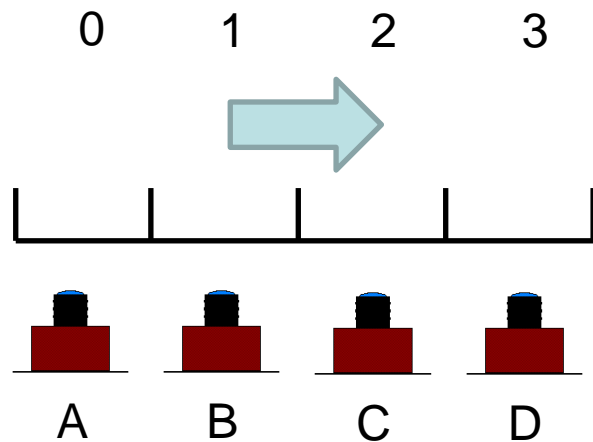


Tabella della verità  
per il caso  $S = 0$   
(priorità riempimento  
a destra)



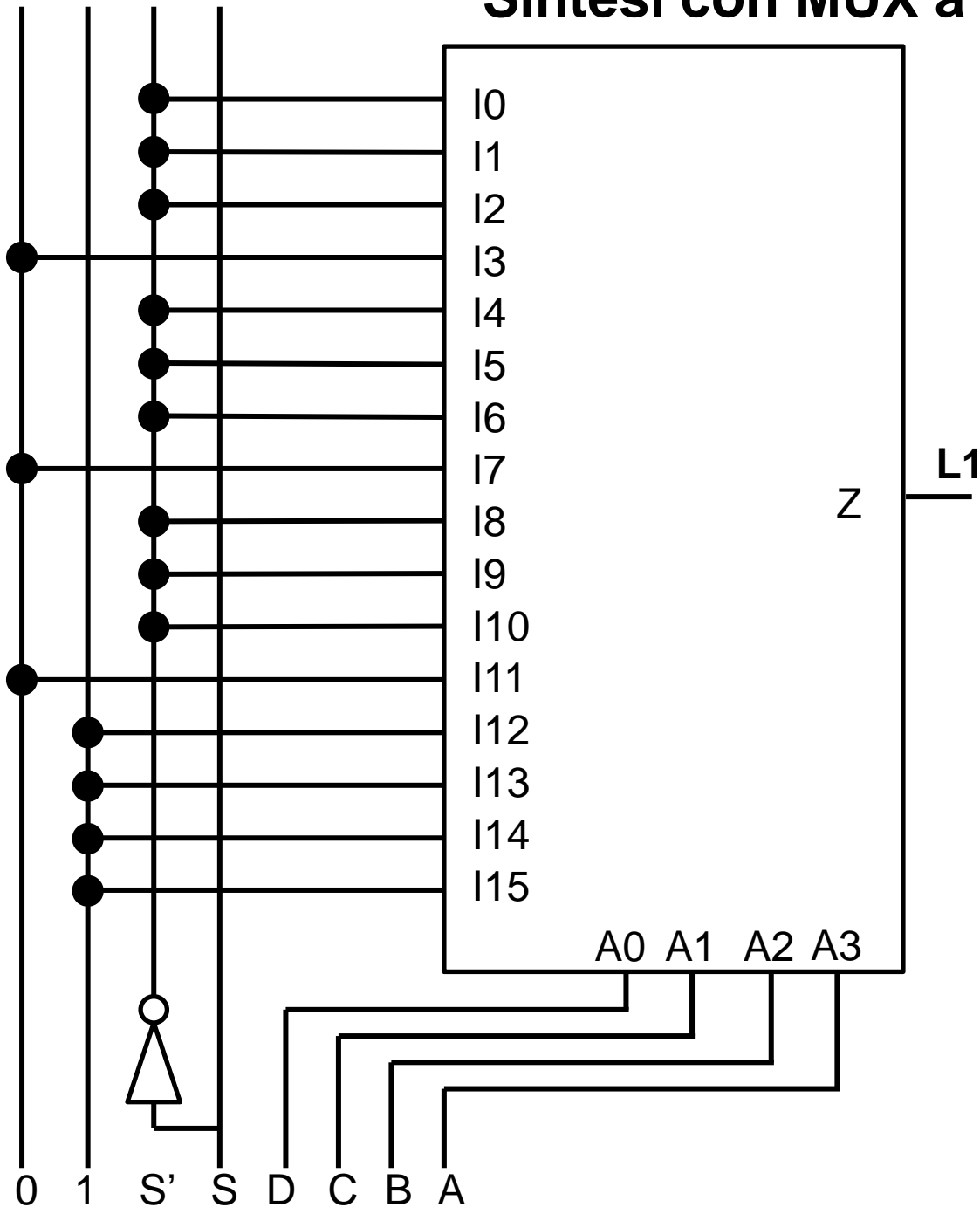
S	A	B	C	D	POS	L1	L2	Z
0	0	0	0	0	3	1	1	0
0	0	0	0	1	2	1	0	
0	0	0	1	0	3	1	1	
0	0	0	1	1	1	0	1	
0	0	1	0	0	3	1	1	
0	0	1	0	1	2	1	0	
0	0	1	1	0	3	1	1	
0	0	1	1	1	0	0	0	
0	1	0	0	0	3	1	1	
0	1	0	0	1	2	1	0	
0	1	0	1	0	3	1	1	
0	1	0	1	1	1	0	1	
0	1	1	0	0	3	1	1	
0	1	1	0	1	2	1	0	
0	1	1	1	0	3	1	1	
0	1	1	1	1	-	-	-	1

Tabella della verità  
per il caso  $S = 1$   
(priorità riempimento  
a sinistra)



S	A	B	C	D	POS	L1	L2	Z
1	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	
1	0	0	1	0	0	0	0	
1	0	0	1	1	0	0	0	
1	0	1	0	0	0	0	0	
1	0	1	0	1	0	0	0	
1	0	1	1	0	0	0	0	
1	0	1	1	1	0	0	0	
1	1	0	0	0	1	0	1	
1	1	0	0	1	1	0	1	
1	1	0	1	0	1	0	1	
1	1	0	1	1	1	0	1	
1	1	1	0	0	2	1	0	
1	1	1	0	1	2	1	0	
1	1	1	1	0	3	1	1	
1	1	1	1	1	-	-	-	1

# Sintesi con MUX a 4 vie

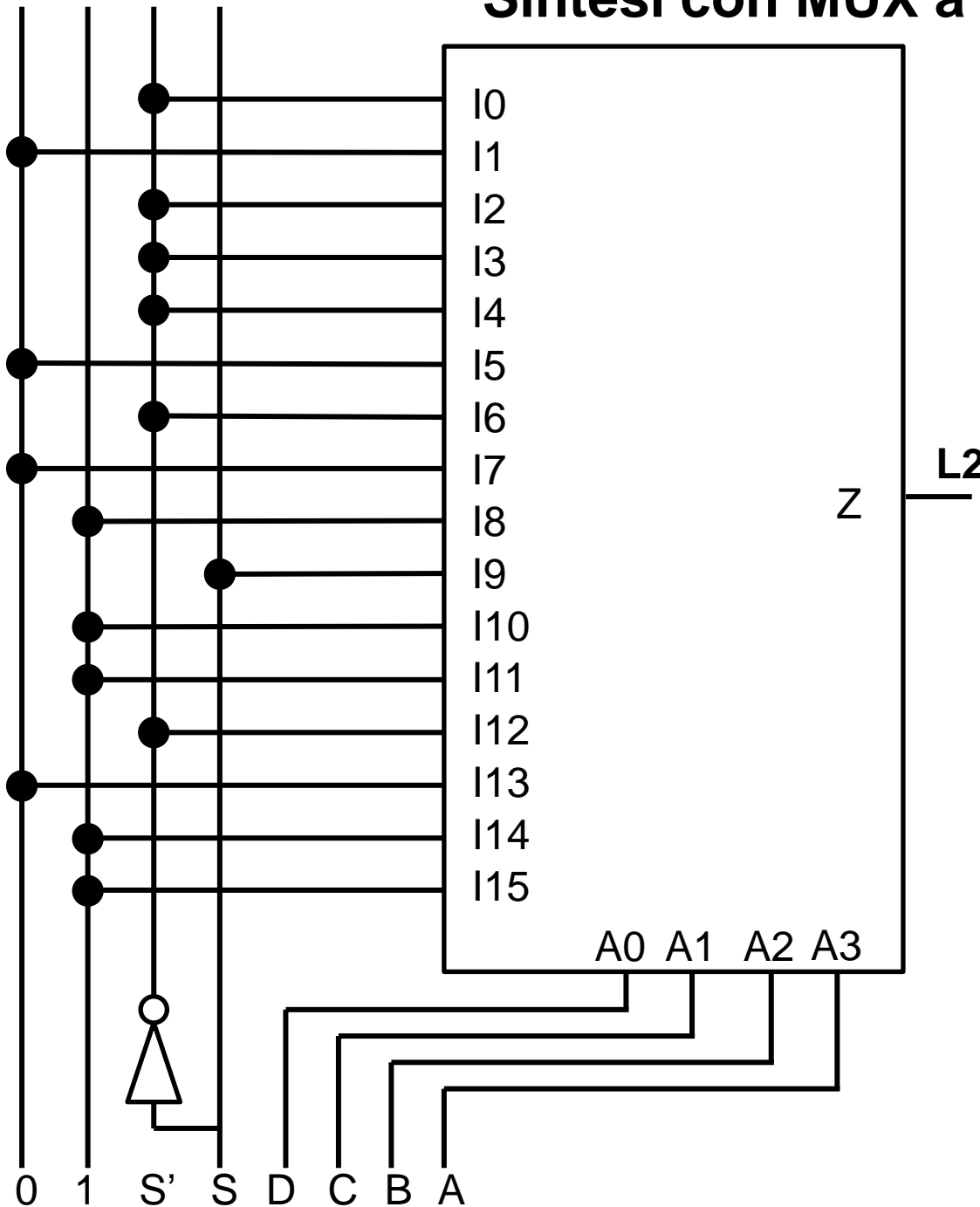


L1

ABCD	S=0	S=1	F(S)
<b>0000</b>	1	0	S'
<b>0001</b>	1	0	S'
<b>0010</b>	1	0	S'
<b>0011</b>	0	0	0
<b>0100</b>	1	0	S'
<b>0101</b>	1	0	S'
<b>0110</b>	1	0	S'
<b>0111</b>	0	0	0
<b>1000</b>	1	0	S'
<b>1001</b>	1	0	S'
<b>1010</b>	1	0	S'
<b>1011</b>	0	0	0
<b>1100</b>	1	1	1
<b>1101</b>	1	1	1
<b>1110</b>	1	1	1
<b>1111</b>	-	-	-

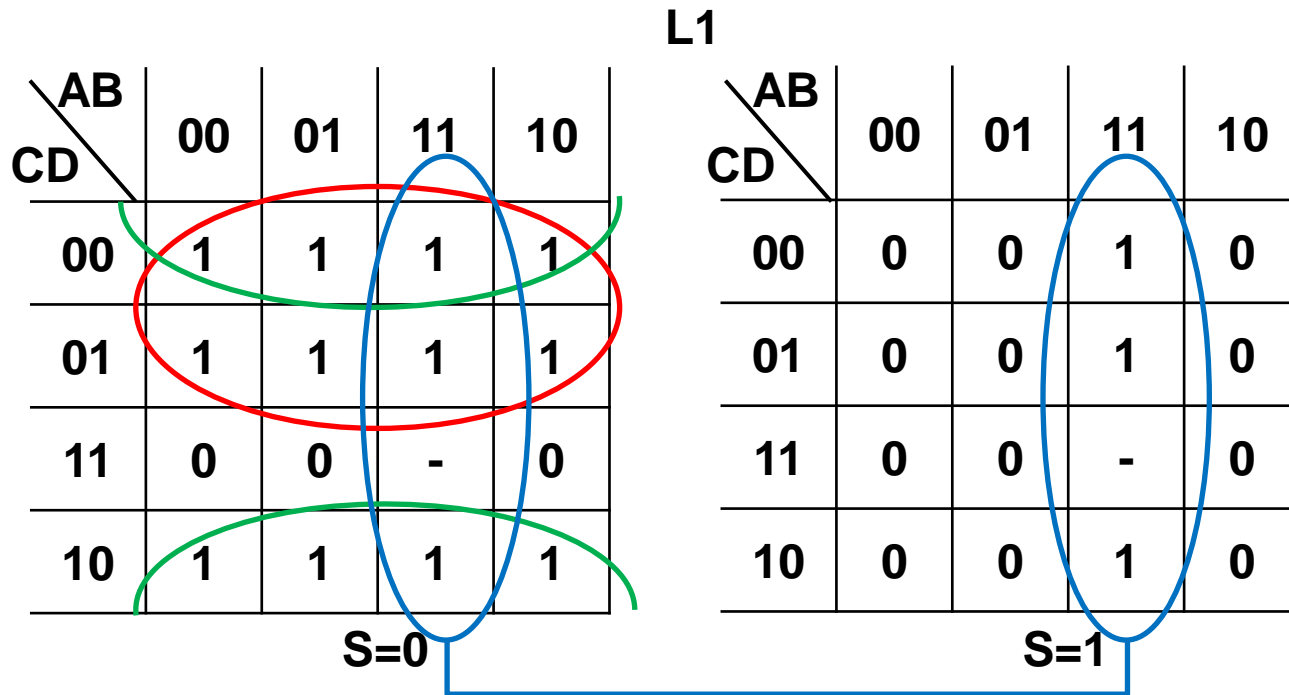
# Sintesi con MUX a 4 vie

L2



ABCD	S=0	S=1	F(S)
0000	1	0	S'
0001	0	0	0
0010	1	0	S'
0011	1	0	S'
0100	1	0	S'
0101	0	0	0
0110	1	0	S'
0111	0	0	0
1000	1	1	1
1001	0	1	S
1010	1	1	1
1011	1	1	1
1100	1	0	S'
1101	0	0	0
1110	1	1	1
1111	-	-	-

# Sintesi Rete di costo minimo SP



$$L1 = S'C' + S'D' + AB$$



# Sintesi Rete di costo minimo SP

L2

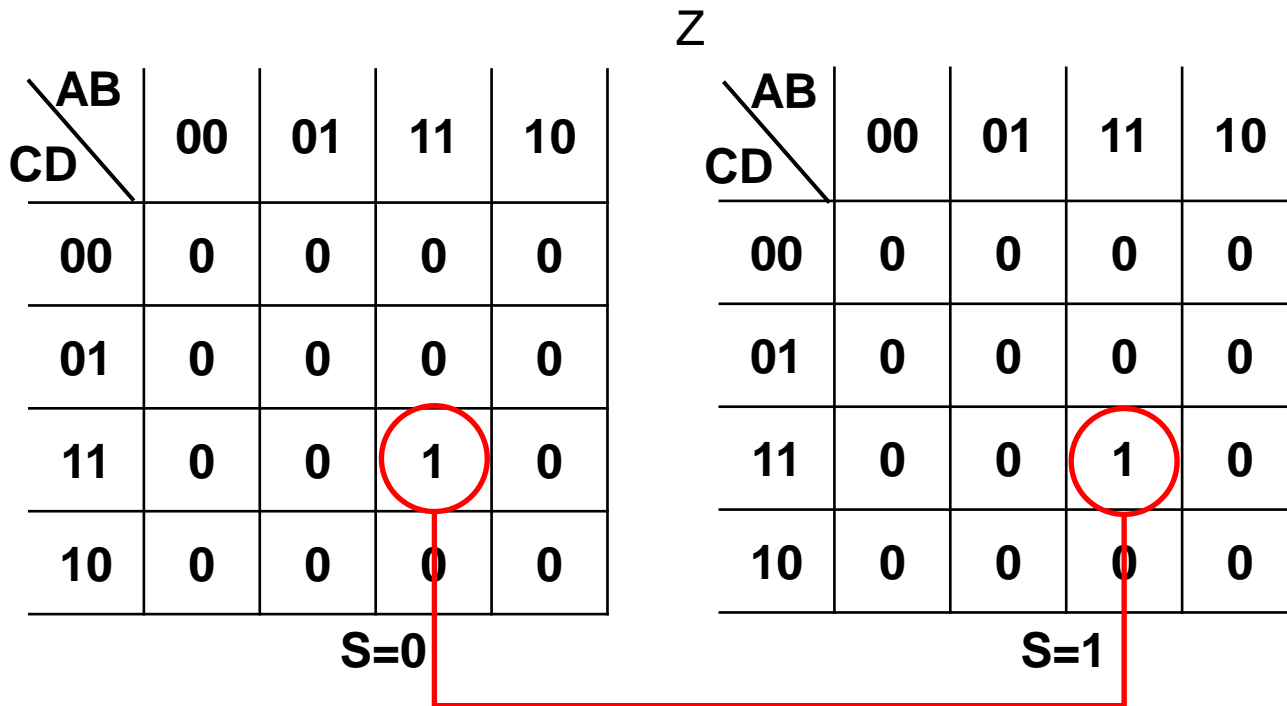
AB \ CD	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	1	0	-	1
10	1	1	1	1

AB \ CD	00	01	11	10
00	0	0	0	1
01	0	0	0	1
11	0	0	-	1
10	0	0	1	1

**S=0**
**S=1**

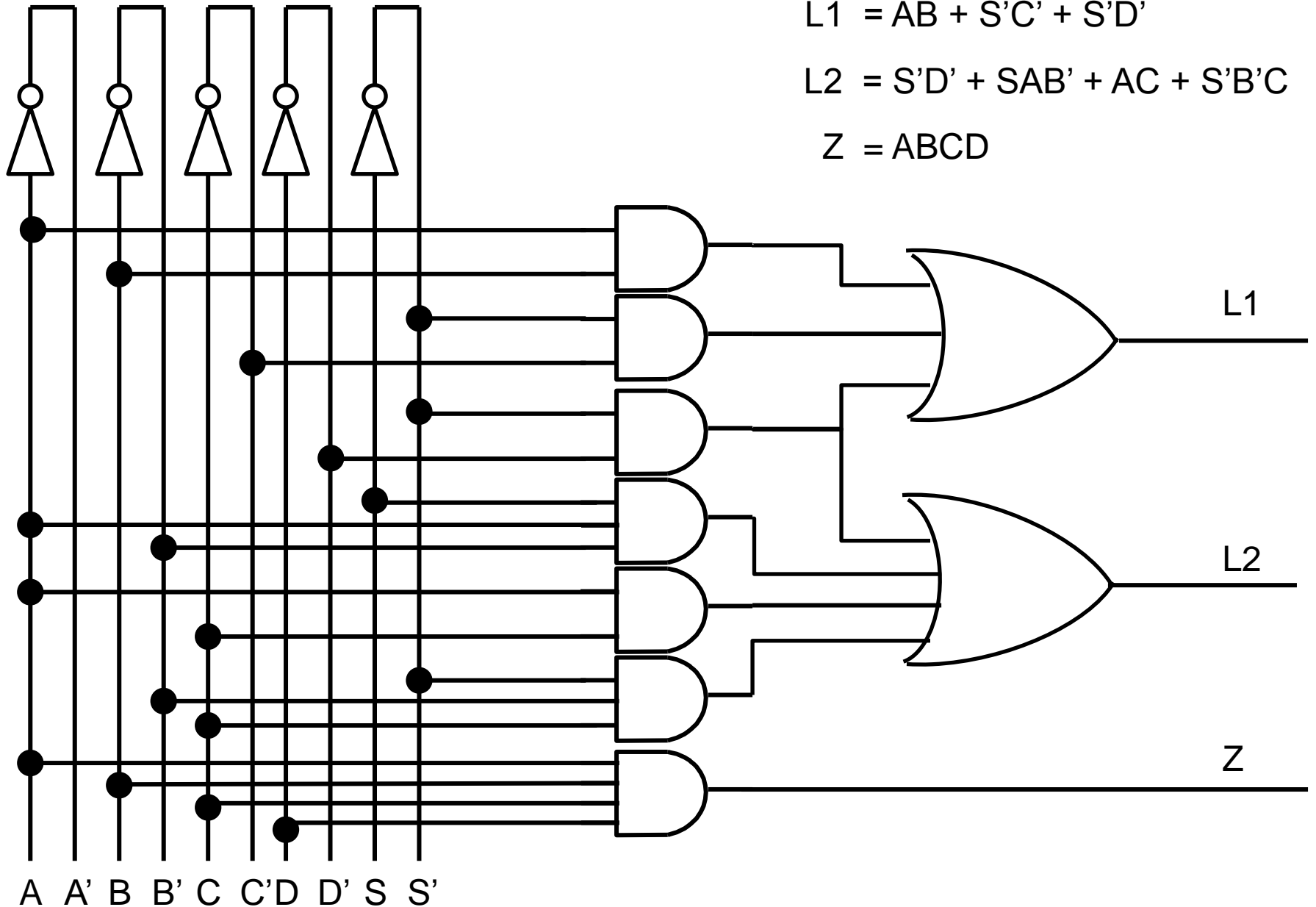
$$L2 = S'D' + SAB' + AC + S'B'C$$

# Sintesi Rete di costo minimo SP



$$Z = ABCD$$

# Sintesi Rete di costo minimo SP



# Sintesi Rete di costo minimo a NAND

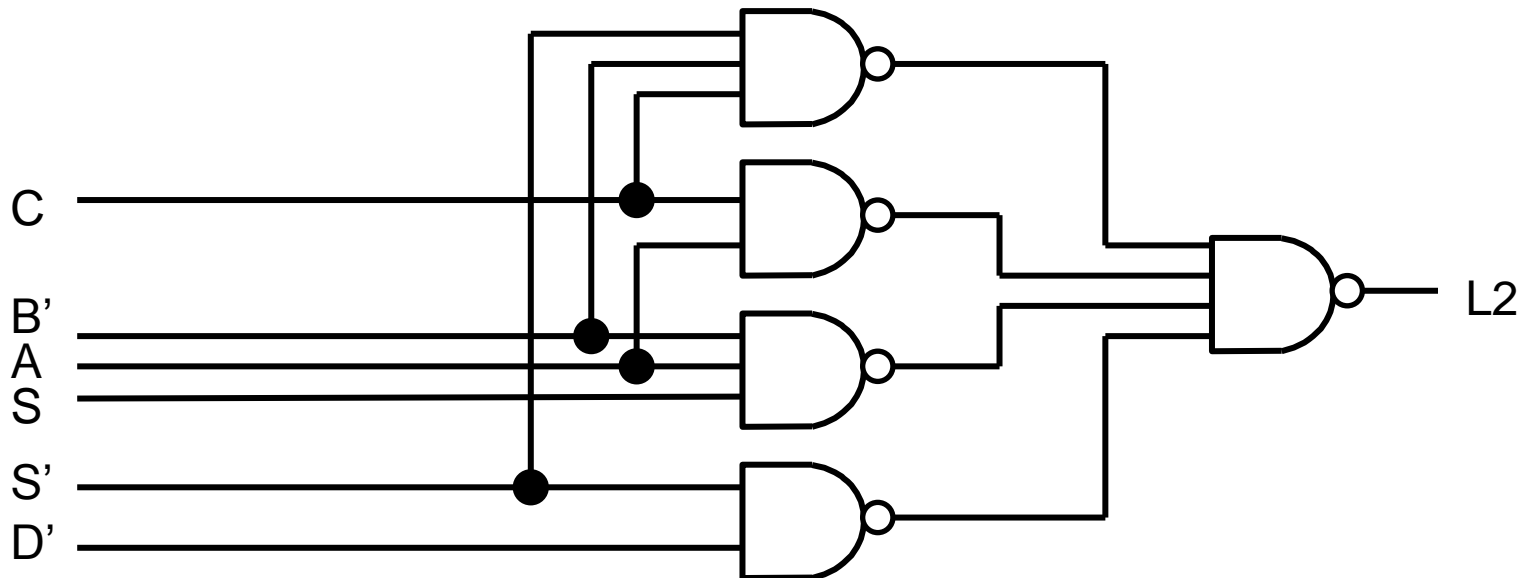
$$L2 = S'D' + SAB' + AC + S'B'C$$

$$L2 = ((S' \cdot D') + (S \cdot A \cdot B') + (A \cdot C) + (S' \cdot B' \cdot C))$$

$$L2 = ((S' \uparrow D') + (S \uparrow A \uparrow B') + (A \uparrow C) + (S' \uparrow B' \uparrow C))$$

$$L2 = ((S' \uparrow D') \uparrow (S \uparrow A \uparrow B') \uparrow (A \uparrow C) \uparrow (S' \uparrow B' \uparrow C))$$

Rete a NAND  
se disponibili  
segnali in  
forma vera e  
negata



# Sintesi Rete di costo minimo a NAND

$$L2 = S'D' + SAB' + AC + S'B'C$$

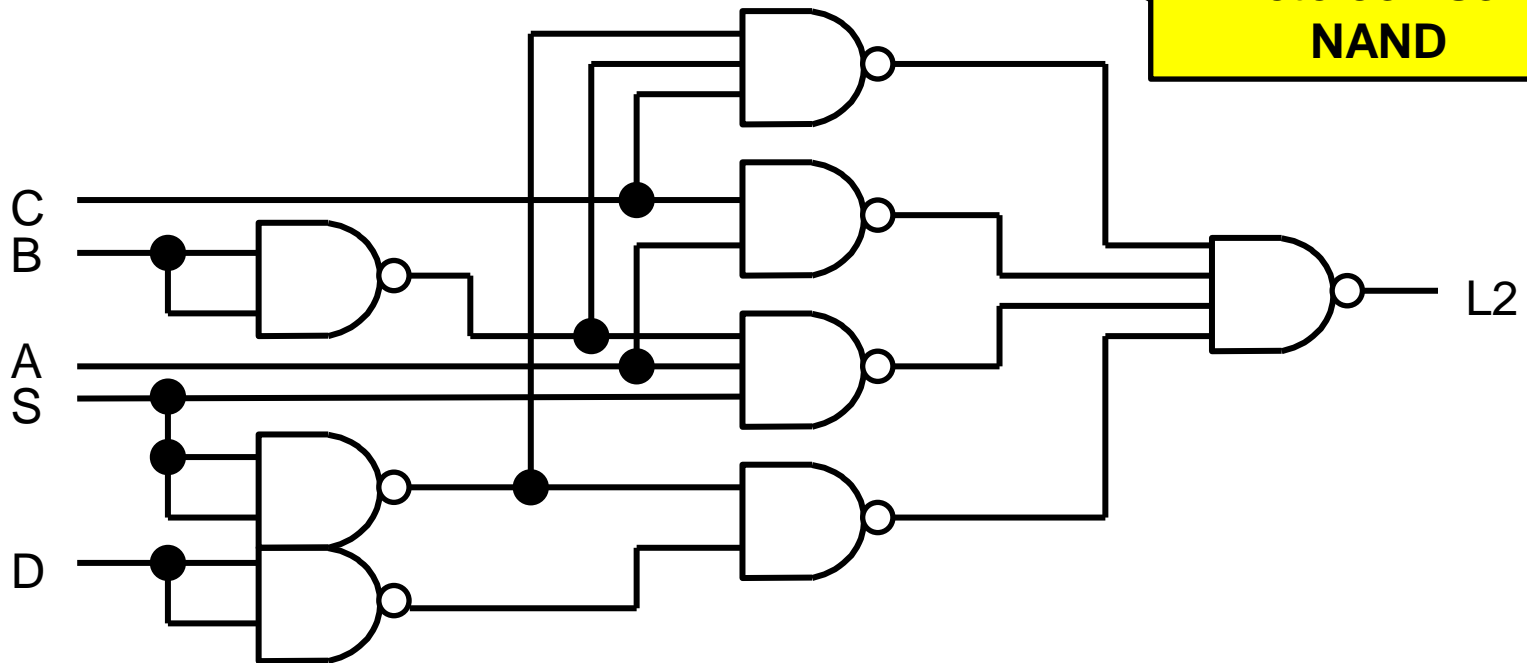
$$L2 = ((S' \cdot D') + (S \cdot A \cdot B') + (A \cdot C) + (S' \cdot B' \cdot C))$$

$$L2 = ((S' \uparrow D') + (S \uparrow A \uparrow B') + (A \uparrow C) + (S' \uparrow B' \uparrow C))$$

$$L2 = ((S' \uparrow D') \uparrow (S \uparrow A \uparrow B') \uparrow (A \uparrow C) \uparrow (S' \uparrow B' \uparrow C))$$

$$L2 = (((S \uparrow S) \uparrow (D \uparrow D)) \uparrow (S \uparrow A \uparrow (B \uparrow B)) \uparrow (A \uparrow C) \uparrow ((S \uparrow S) \uparrow (B \uparrow B) \uparrow C))$$

Rete a NAND  
se disponibili  
segnali in  
forma vera e  
negata



Rete con soli  
NAND

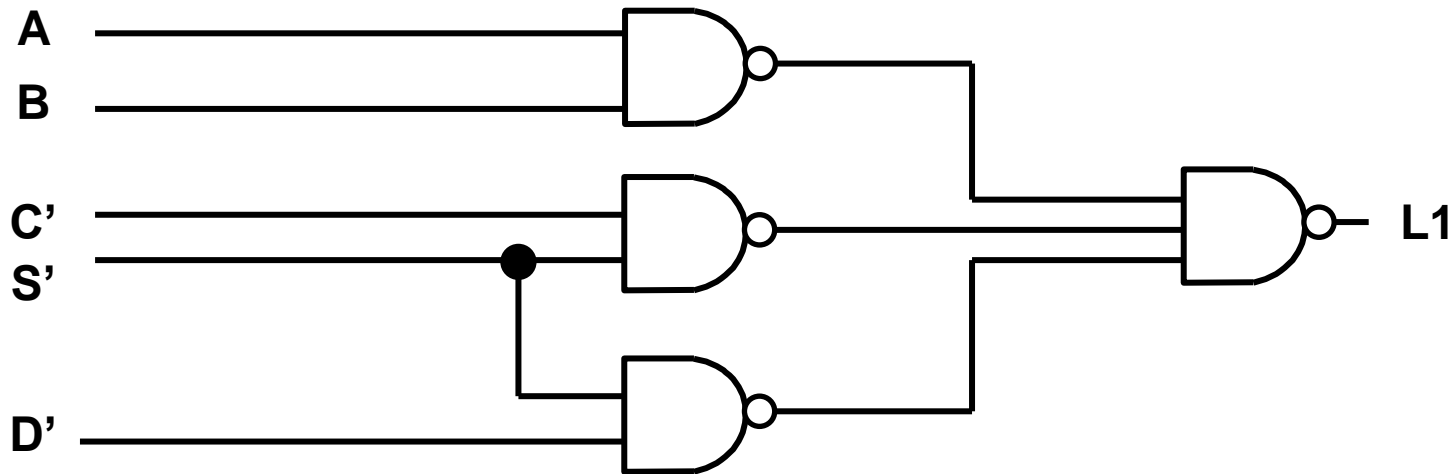
# Sintesi Rete di costo minimo a NAND

$$L1 = AB + S'C' + S'D'$$

$$L1 = (A \cdot B) + (S' \cdot C') + (S' \cdot D')$$

$$L1 = (A \uparrow B) + (S' \uparrow C') + (S' \uparrow D')$$

$$L1 = (A \uparrow B) \uparrow (S' \uparrow C') \uparrow (S' \uparrow D')$$



# Sintesi Rete di costo minimo a NAND

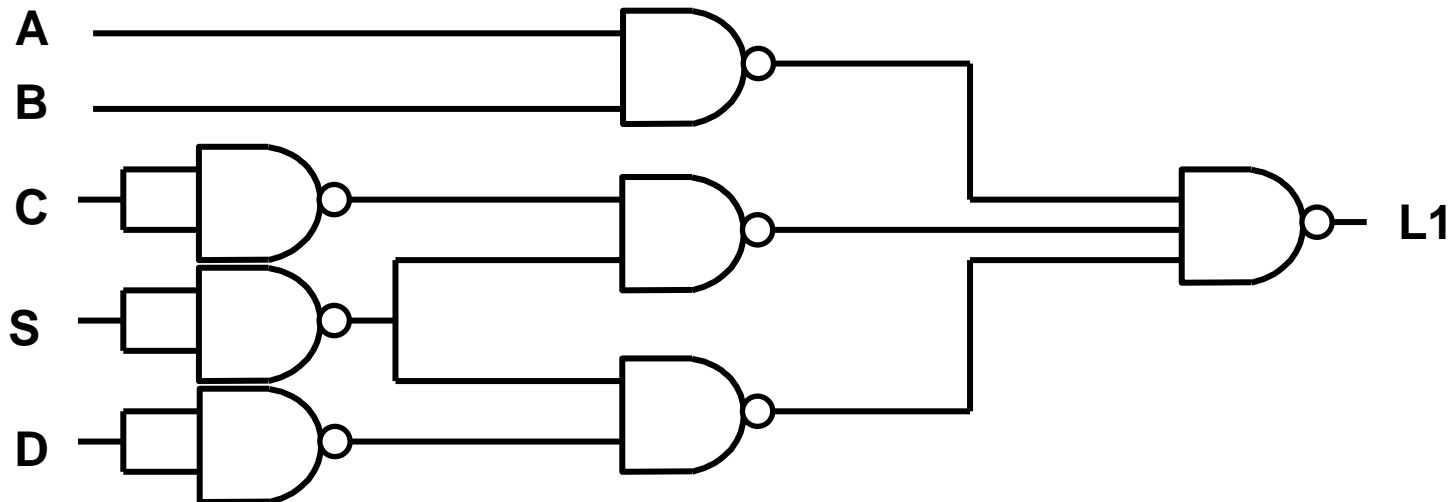
$$L1 = AB + S'C' + S'D'$$

$$L1 = (A \cdot B) + (S' \cdot C') + (S' \cdot D')$$

$$L1 = (A \uparrow B) + (S' \uparrow C') + (S' \uparrow D')$$

$$L1 = (A \uparrow B) \uparrow (S' \uparrow C') \uparrow (S' \uparrow D')$$

$$L1 = (A \uparrow B) \uparrow ((S \uparrow S) \uparrow (C \uparrow C)) \uparrow ((S \uparrow S) \uparrow (D \uparrow D))$$



# Sintesi Rete di costo minimo a NAND

Posso anche manipolarla per ottenere un'espressione senza segnali negati, facendo in modo che i segnali negati compaiano solo come operandi di un OR

$$L1 = AB + S'C' + S'D'$$

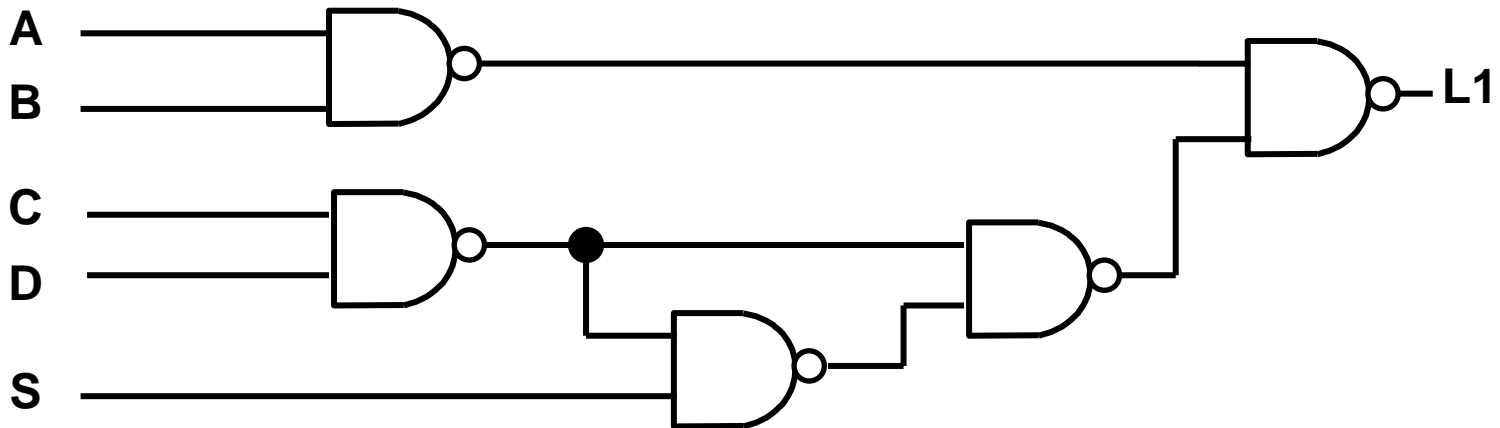
$$L1 = AB + S'(C' + D') + (C' + D')(C' + D)'$$

$$L1 = AB + (C'+D')(S' + (C' + D)')$$

$$L1 = AB + (C'+D')(S' + CD) \quad \text{SPSP}$$

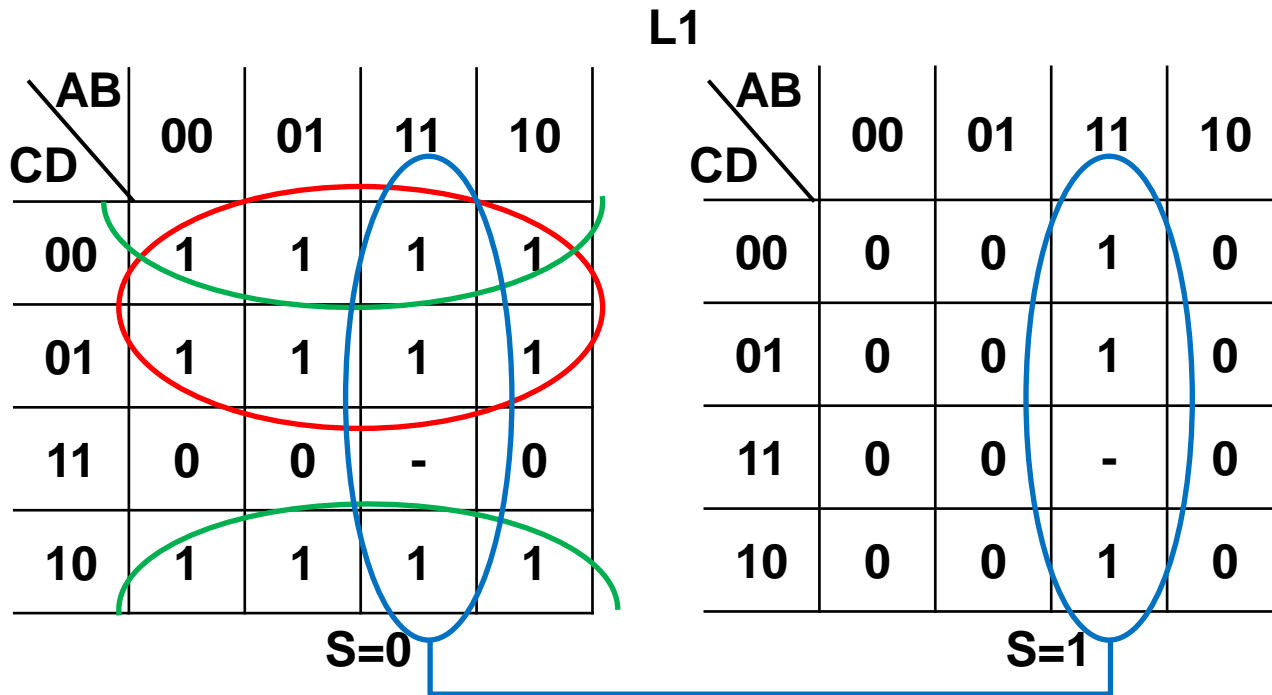
$$L1 = (A \cdot B) + ((C' + D') \cdot (S' + (C \cdot D)))$$

$$L1 = (A \uparrow B) \uparrow ((C \uparrow D) \uparrow (S \uparrow (C \uparrow D)))$$





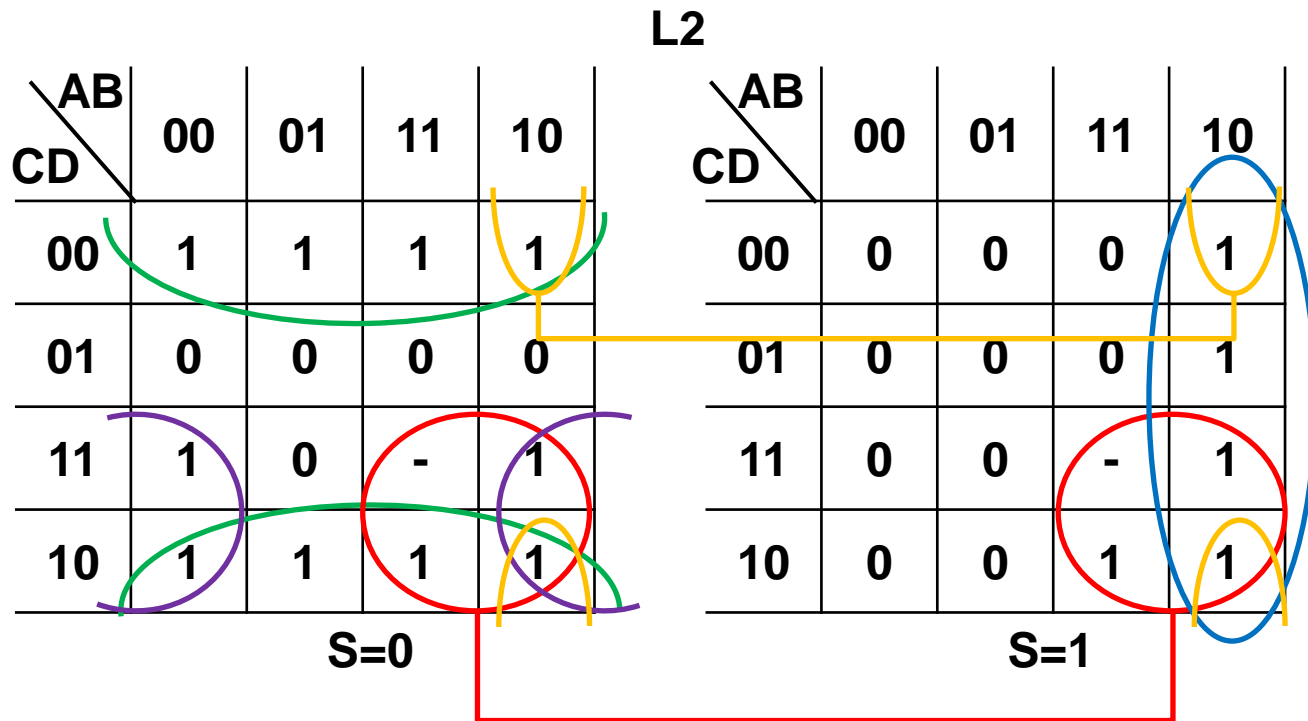
# Rimozione Alee Statiche



$$L1 = S'C' + S'D' + AB$$

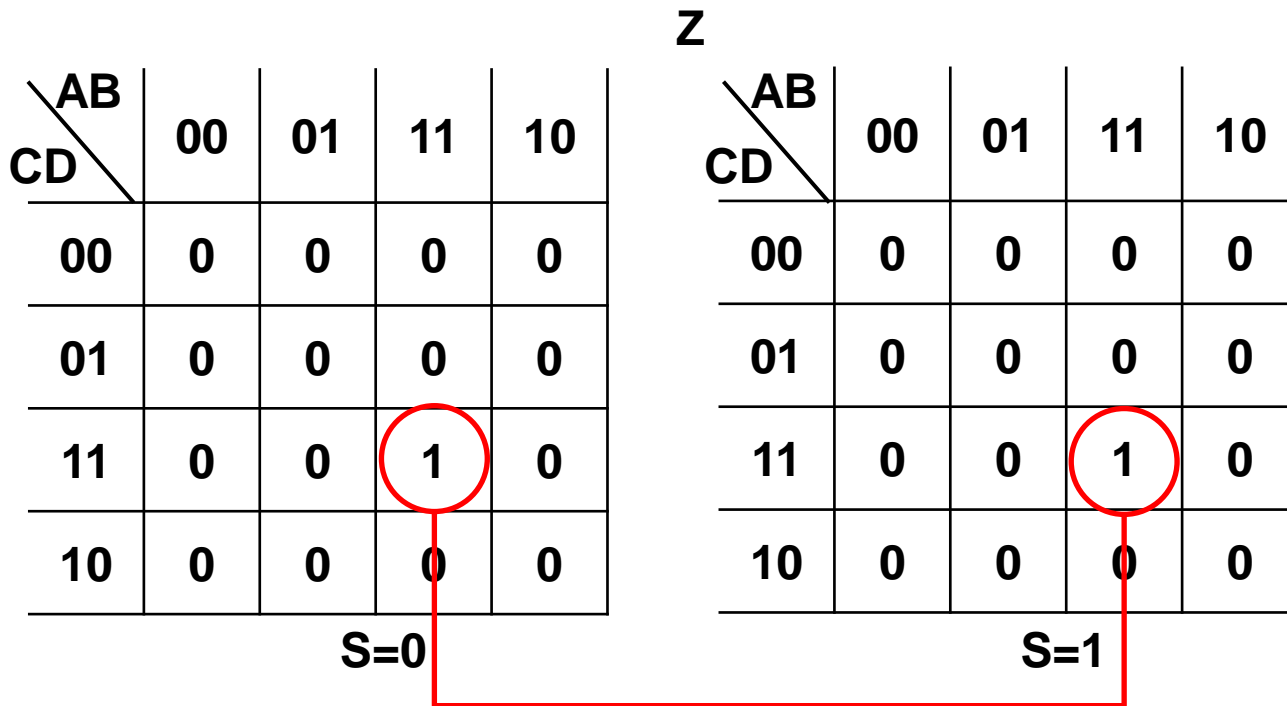


# Rimozione Alee Statiche



$$L2 = S'D' + SAB' + AC + S'B'C + AB'D'$$

# Rimozione Alee Statiche

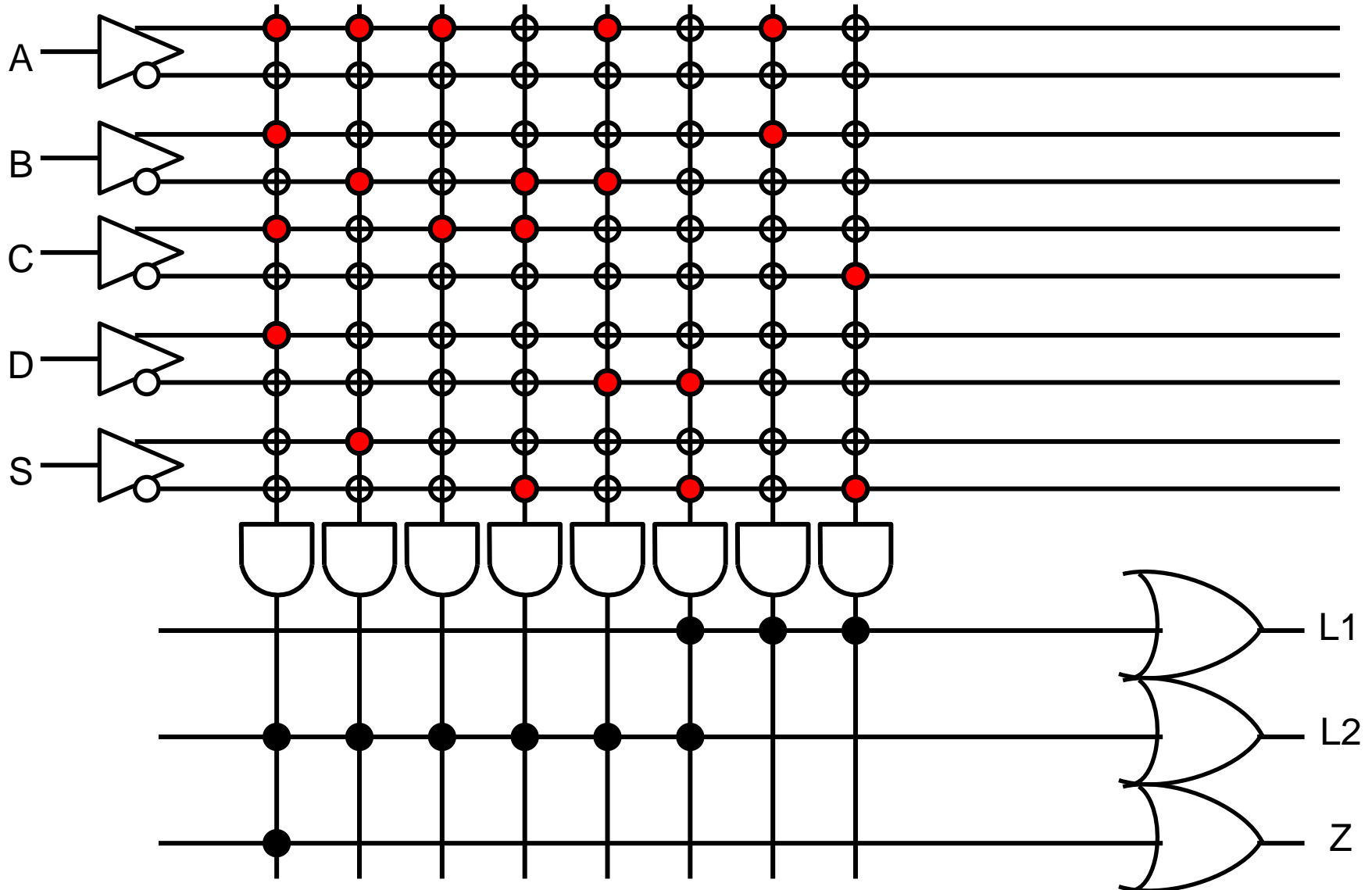


$$Z = ABCD$$



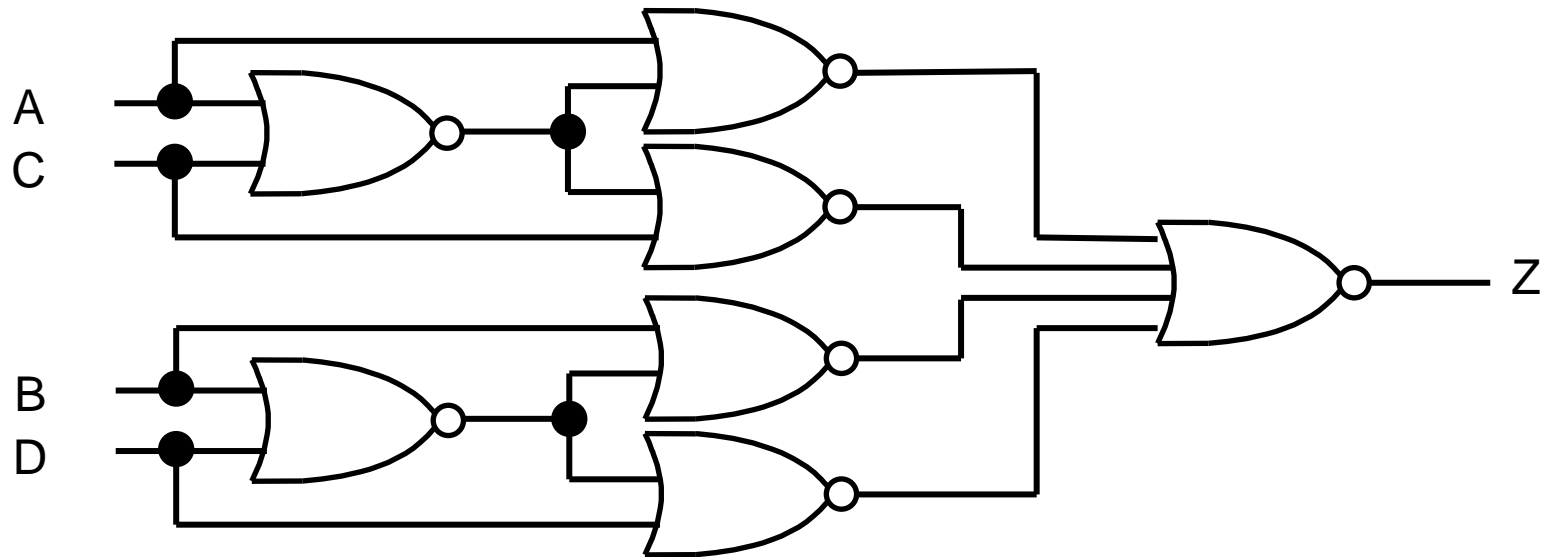
# Sintesi con PAL di reti senza alee

$$L1 = AB + S'C' + S'D' \quad L2 = S'D' + SAB' + AC + S'B'C + A B' D' \quad Z = ABCD$$

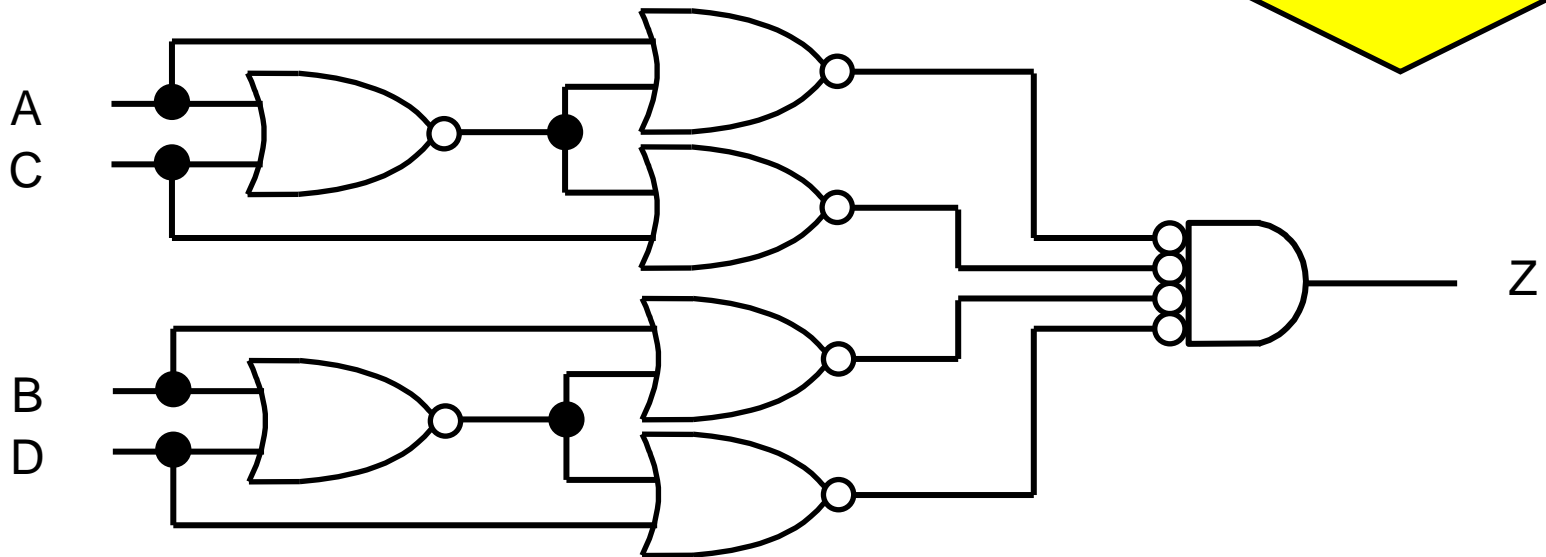
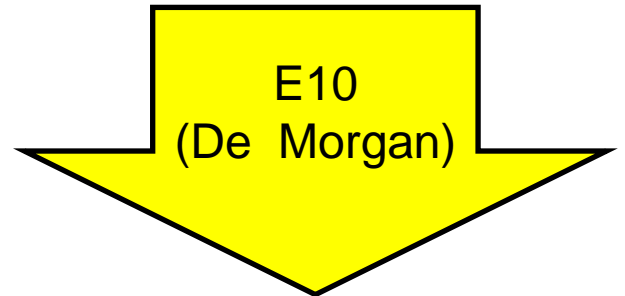
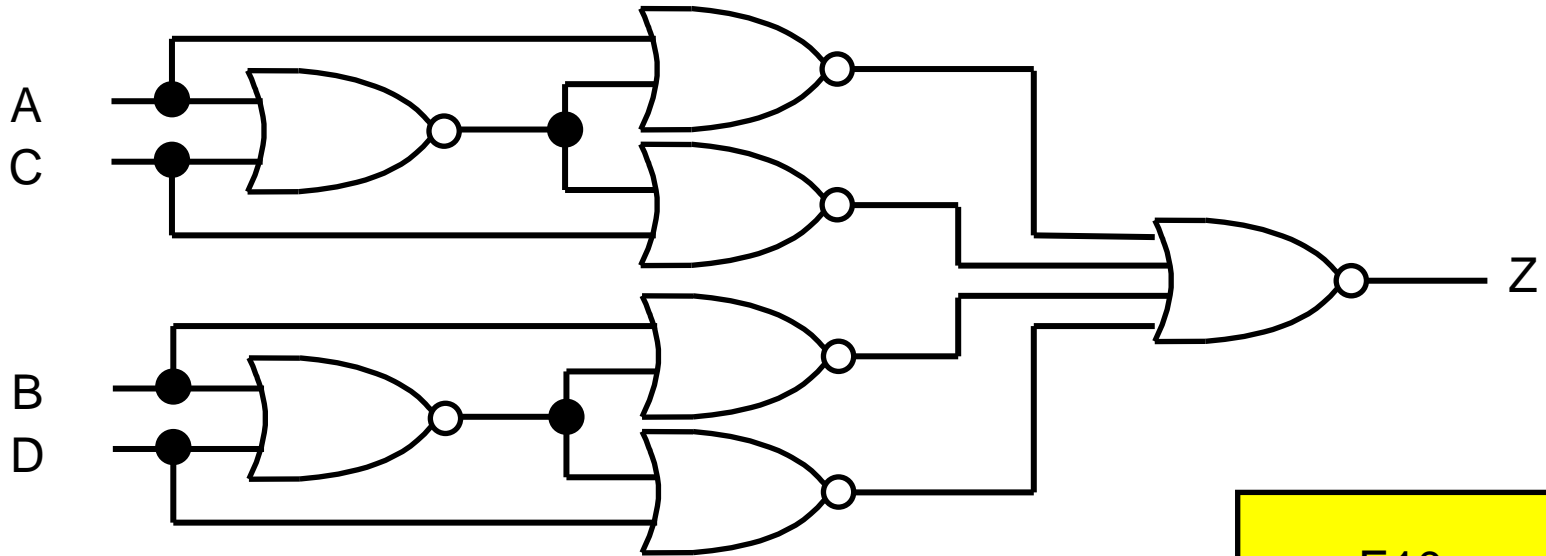


# Esercizio Analisi RC

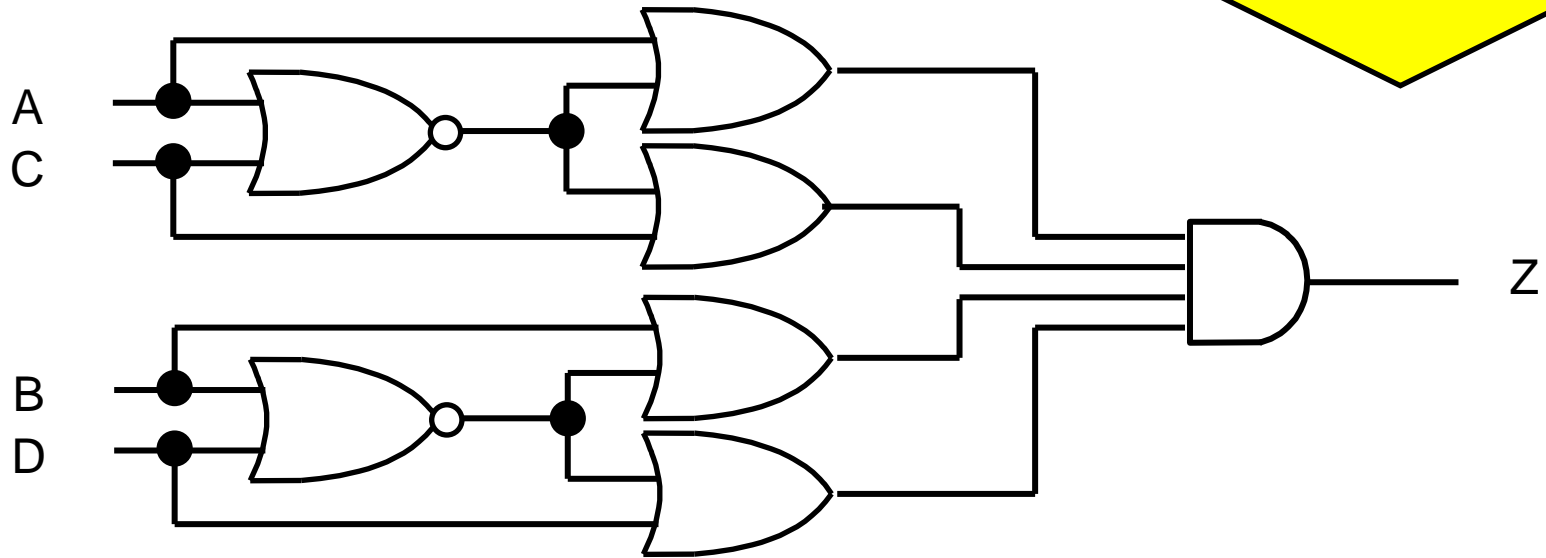
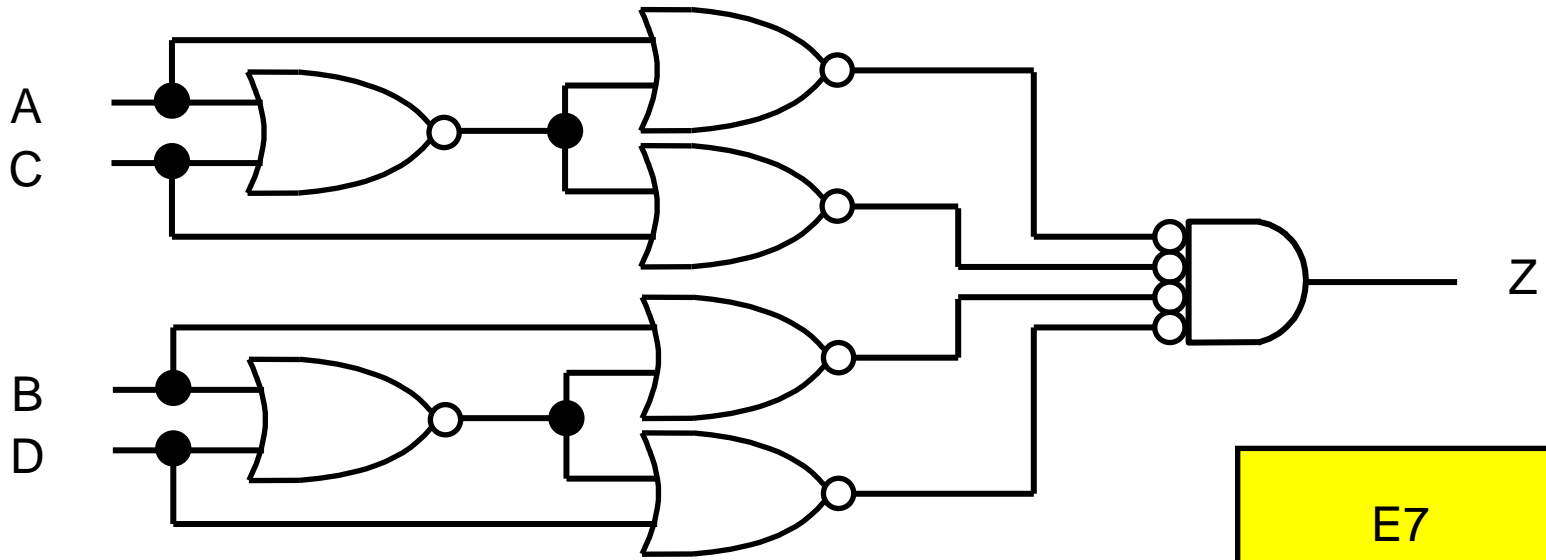
- ◆ Data la rete a NOR seguente, analizzarla per via grafica per determinare una rete equivalente espressa nell'algebra di commutazione.
- ◆ Data l'espressione della rete equivalente, usare i teoremi di equivalenza per ridurla alla forma normale PS
- ◆ Dato la forma PS equivalente, definire le mappe di Karnaugh corrispondenti
- ◆ A partire dalle mappe, dare una descrizione del comportamento della rete.



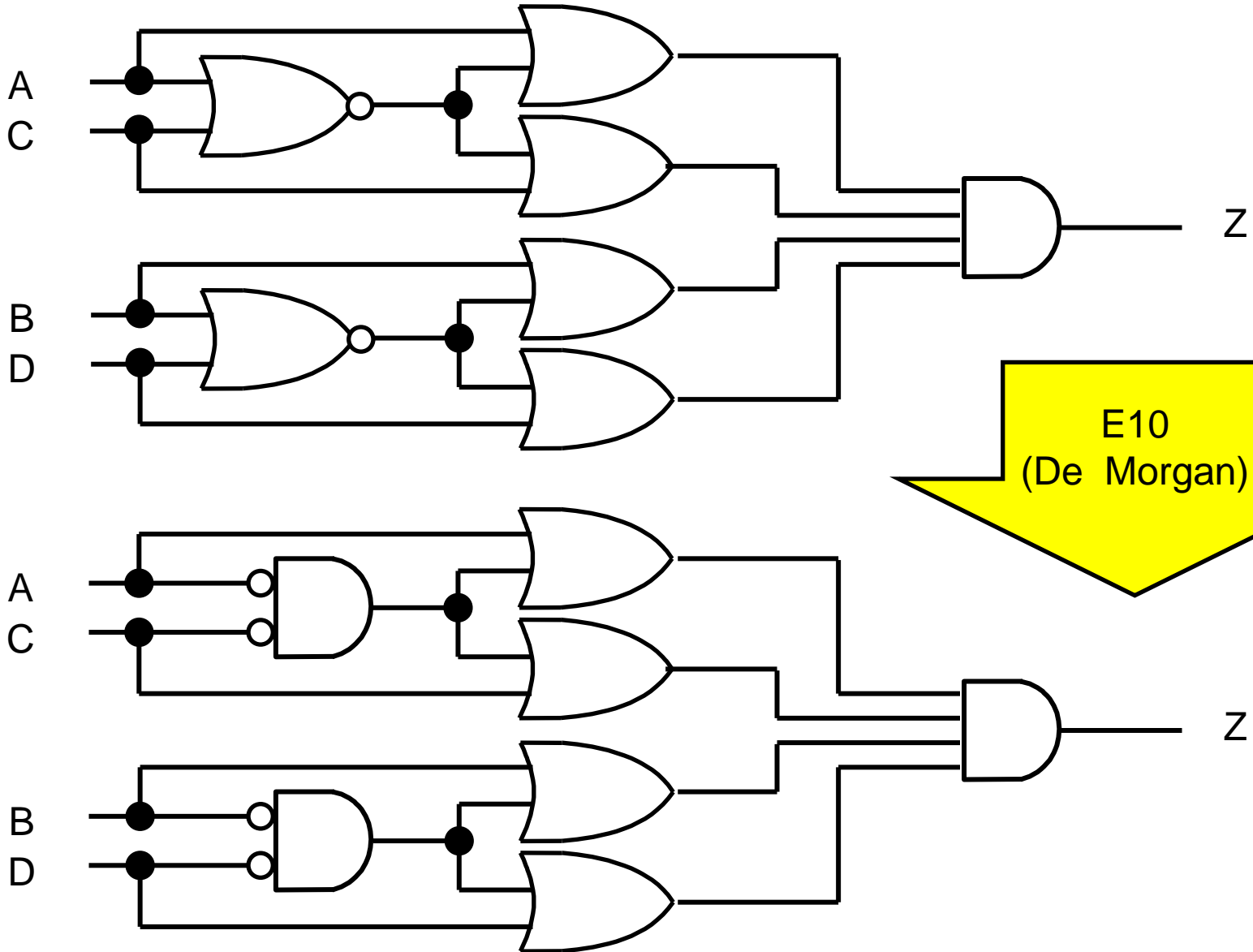
# Analisi per via grafica



# Analisi per via grafica

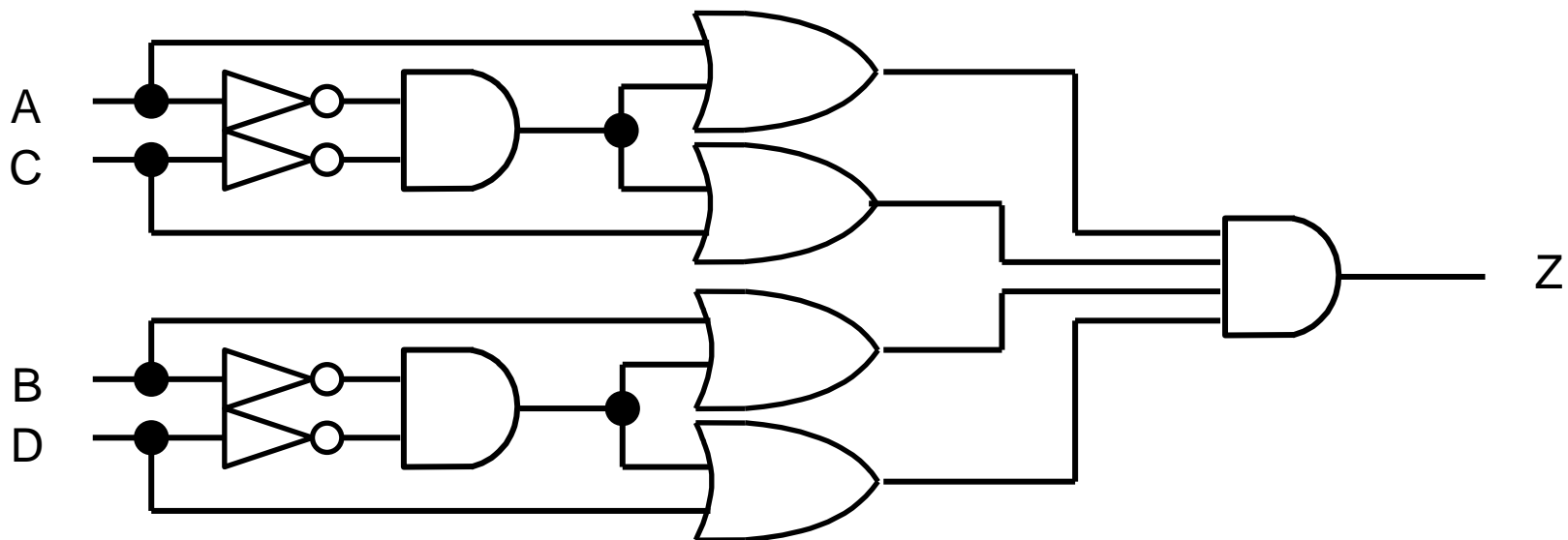
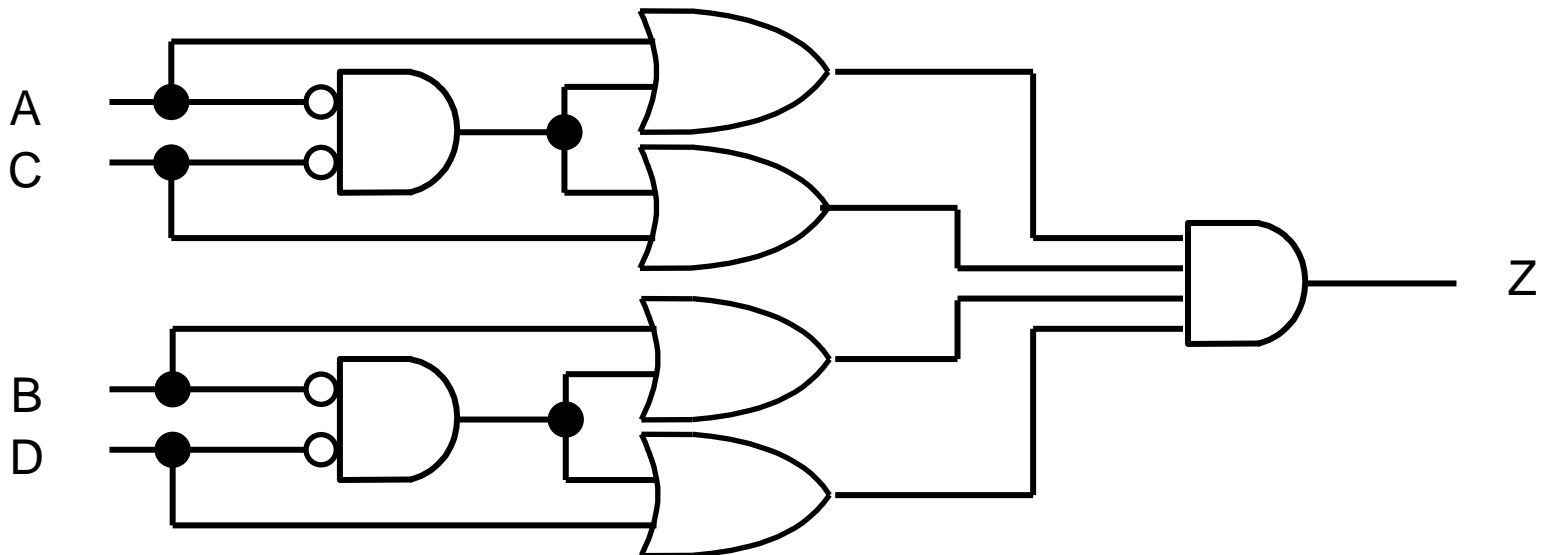


# Analisi per via grafica

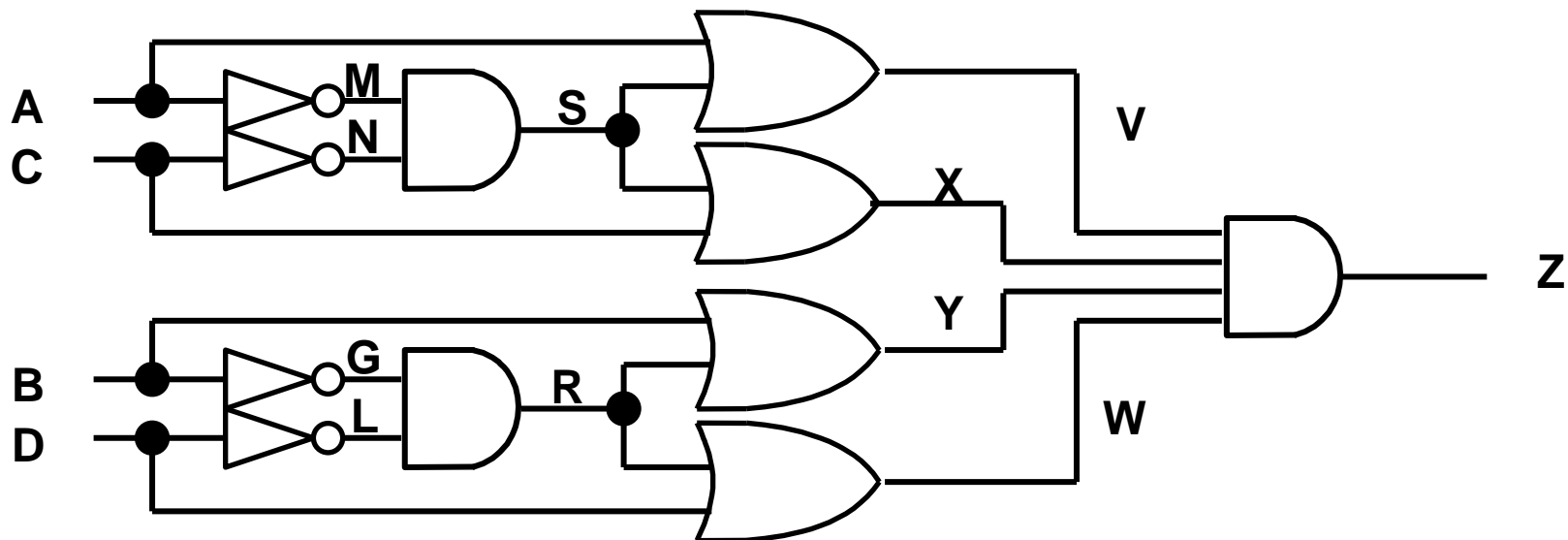




# Analisi per via grafica



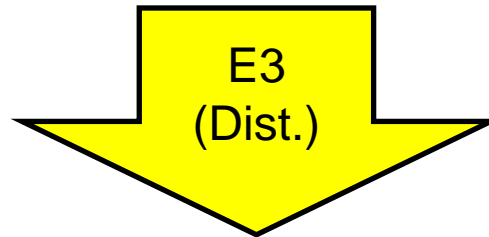
# Espressione



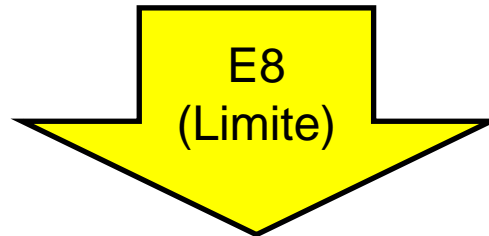
$$\begin{aligned}
 Z &= V X Y W \\
 &= (A+S) (S+C) (B+R) (R+D) \\
 &= (A + MN) (MN + C) (B + GL) (GL + D) \\
 &= (A + A'C') (A'C' + C) (B + B'D') (B'D' + D)
 \end{aligned}$$

# Forma normale PS

$$Z = (A + A'C') (A'C' + C) (B + B'D') (B'D' + D)$$



$$Z = (A + A')(A + C') (A' + C)(C' + C) (B + B')(B + D') (B' + D)(D' + D)$$



$$Z = (A + C') (A' + C) (B + D') (B' + D)$$

# Mappe di Karnaugh

$$Z = (A + C') (A' + C) (B + D') (B' + D)$$

$$(A + C') (A' + C) (B + D') (B' + D)$$

AB \ CD	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

