

# Tempistiche e diagrammi temporali

**La funzione è giusta ma il circuito non funziona**

# Perché

---

- ▶ **I circuiti che costruiamo hanno ingressi che variano nel tempo**
  - ▶ Perché qualcuno preme un pulsante
  - ▶ Perché cambia la lettura di un sensore
  - ▶ Perché i dati sono in forma seriale
  - ▶ Etc.
- ▶ **Anche le uscite cambieranno nel tempo**
  - ▶ Passato il transitorio, si stabilizzano sul valore dettato dalla funzione logica
- ▶ **Come influisce la rete sulle tempistiche delle uscite?**
  - ▶ Ogni porta logica introduce un ritardo
  - ▶ La somma dei ritardi lungo il cammino del segnale definisce il ritardo finale dell'uscita
  - ▶ Ci interessa poter rappresentare il comportamento del circuito durante la fase di transitorio

# Diagramma temporale

---

- ▶ **Rappresentazione dell'evoluzione dei valori dei nodi della rete nel tempo**
  - ▶ Esattamente come i diagrammi temporali in elettronica analogica
  - ▶ Normalmente però non ci interessano i dettagli della fase di salita e discesa delle singole porte
  - ▶ Usiamo solo due livelli
- ▶ **Due versioni**
  - ▶ **Senza ritardi delle porte:** le variazioni degli ingressi si propagano immediatamente ai nodi interni e all'uscita
    - ▶ Utile per verificare e debuggare la funzione logica
  - ▶ **Con i ritardi delle porte:** le variazioni degli ingressi impiegano del tempo per propagarsi ai nodi interni e all'uscita
    - ▶ Utile per valutare prestazioni, trovare i percorsi critici ed analizzare eventuali anomalie

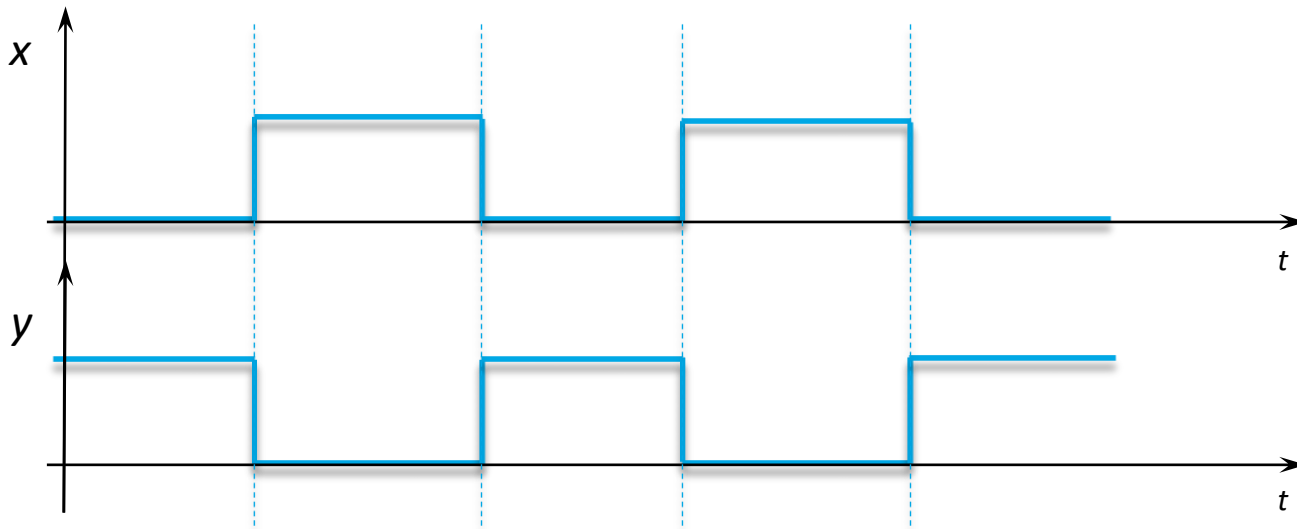
# Diagramma dell'invertitore

---

## ► Due nodi



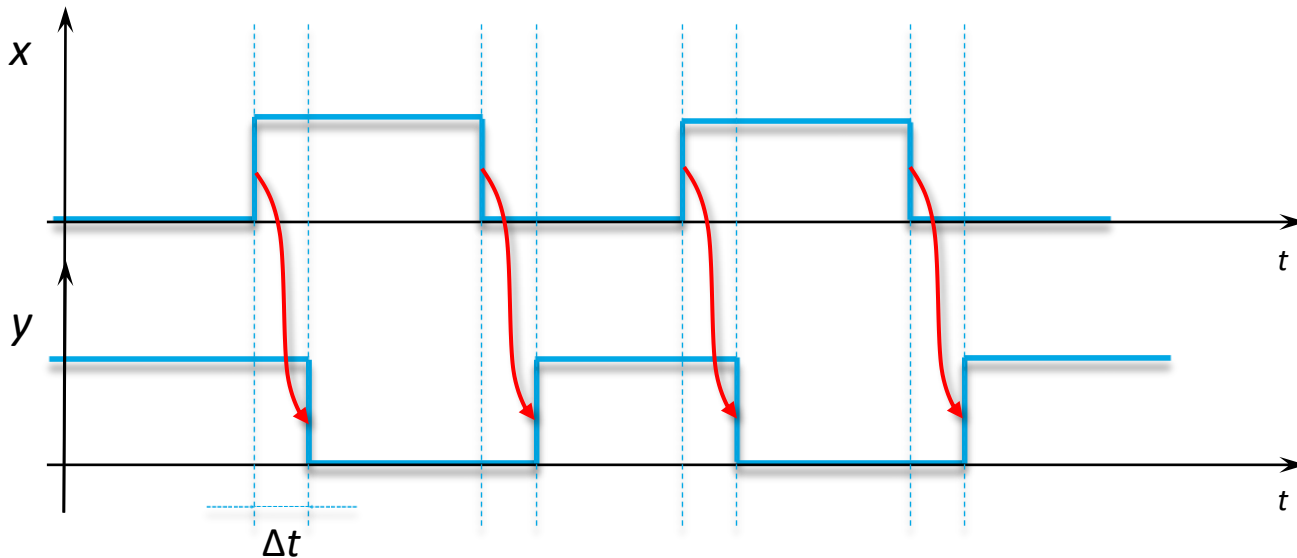
- Ingresso  $x$  e uscita  $y$
- No ritardo attraverso la porta
- Per ogni istante di tempo, l'uscita ha valore opposto a quello dell'ingresso



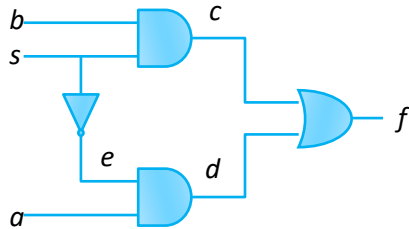
# Diagramma dell'invertitore con ritardo

## ► Due nodi

- Ingresso  $x$  e uscita  $y$
- Ritardo  $\Delta t$  attraverso la porta
- L'uscita raggiunge il valore finale solo dopo il ritardo della porta

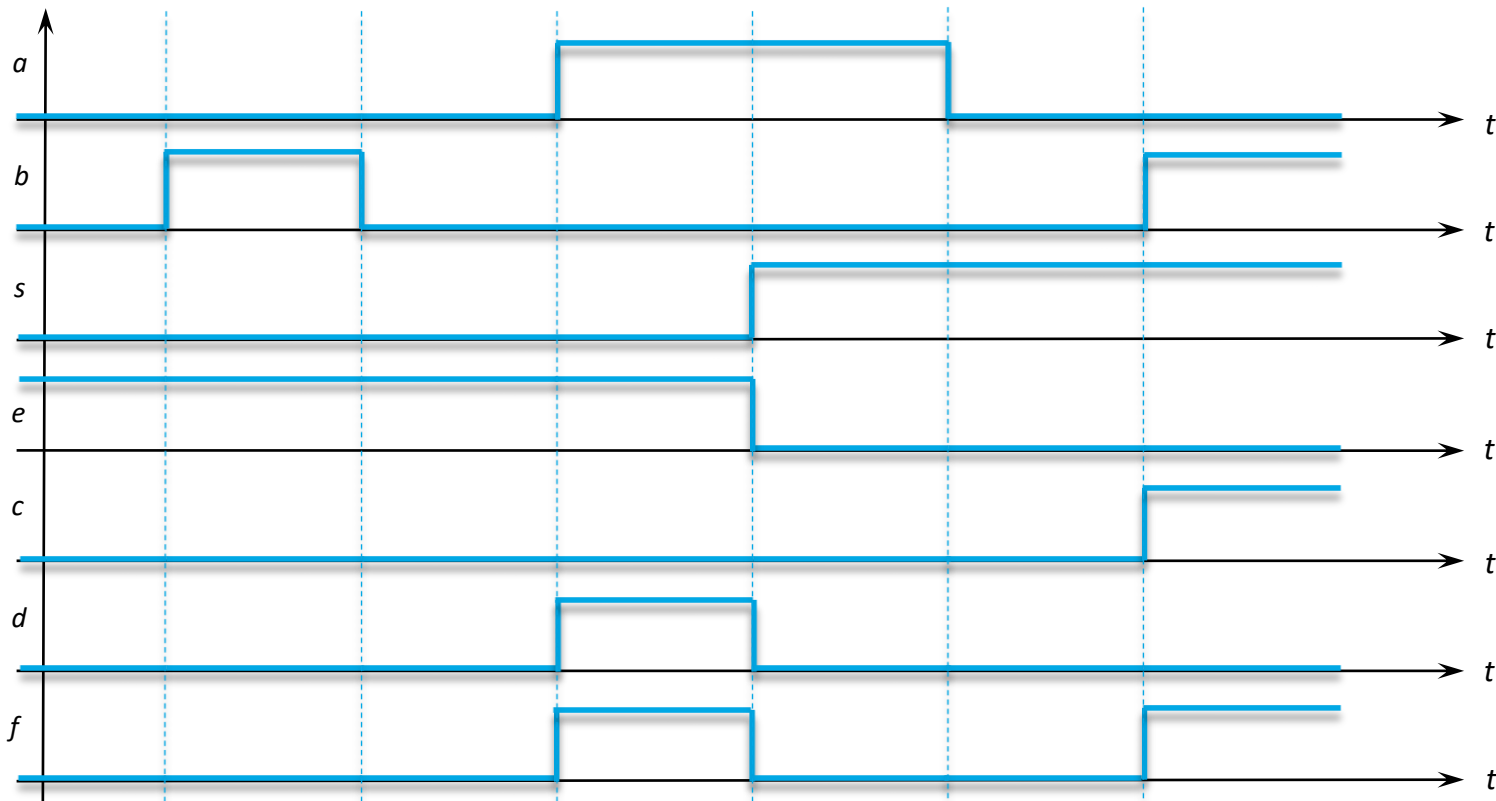


# Rete logica: il multiplexer



## Passi

- Identifichiamo i nodi interni
- Definiamo degli stimoli di ingresso
- Determiniamo i valore dei restanti nodi

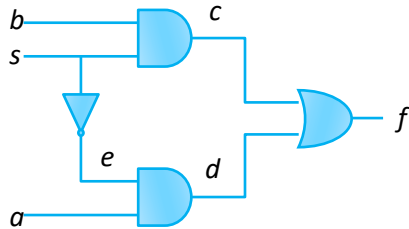


# Come completare un diagramma

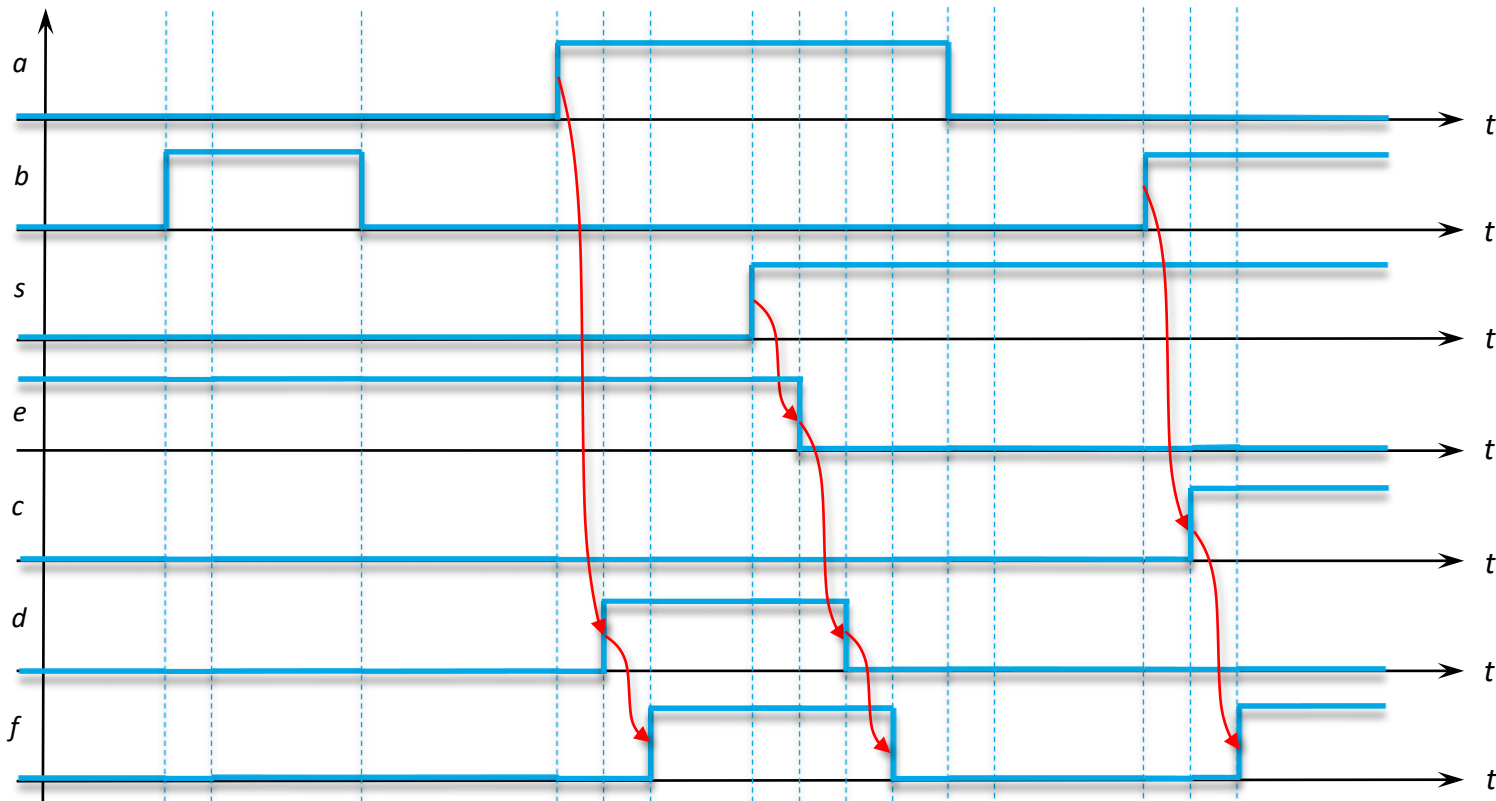
---

- ▶ **Si osservi che, se non succede niente, i valori dei nodi non cambiano**
  - ▶ Si assume, per convenzione, che gli ingressi siano costanti per  $t < 0$
  - ▶ Ci interessano solo i punti in cui qualche segnale cambia di valore
- ▶ **Si valuta il valore di tutti i nodi al tempo  $t = 0$  applicando la funzione logica a partire dagli ingressi**
  - ▶ Si assume che a  $t = 0$  tutto sia a regime
- ▶ **Ci si sposta al minimo tempo  $t^*$  per il quale uno qualunque dei segnali (non solo gli ingressi) commuta di valore**
  - ▶ Si dice che il segnale ha un **evento** a  $t^*$
  - ▶ Si calcolano i nuovi valori nella rete
  - ▶ Si fanno commutare i segnali corrispondenti ai nodi che cambiano valore aggiungendo nuovi eventi al diagramma
  - ▶ Tenendo eventualmente conto dei ritardi
- ▶ **Quindi ci si sposta al prossimo evento**
  - ▶ Si ripete la procedura fino ad aver esaurito tutti gli eventi

# Rete logica: il multiplexer con ritardi



- Supponiamo, per semplicità, che le porte abbiano tutte lo stesso ritardo
  - Trasliamo verso destra i segnali all'uscita delle varie porte
  - Accumuliamo i ritardi della rete





# Osservazioni

---

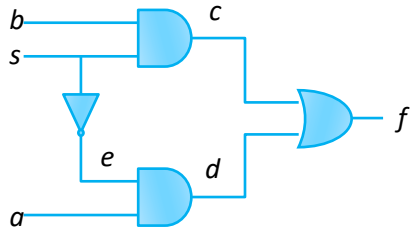
- ▶ **Il ritardo tra gli ingressi e l'uscita non è sempre lo stesso**
  - ▶ Dipende da quale ingresso cambia
  - ▶ Per esempio la variazione su  $a$  impiega due  $\Delta t$  per arrivare all'uscita
  - ▶ La transizione su  $s$  impiega tre  $\Delta t$
  - ▶ Per trovare il massimo e il minimo occorrerebbe provare tutte le possibili transizioni
  - ▶ Ma ci sono metodi più semplici che analizzano la topologia della rete
- ▶ **Modelli di ritardo**
  - ▶ Abbiamo usato un modello di ritardo molto semplice
  - ▶ Lo si può complicare a piacere
    - ▶ Porte con ritardi diversi a seconda di quale ingresso cambia
    - ▶ Aggiustamento dei ritardi secondo il carico (fan-out)
    - ▶ Tenuta in conto dei tempi di salita e discesa
    - ▶ Modello inerziale che elimina brevi transizioni degli ingressi

# Alee (glitch)

---

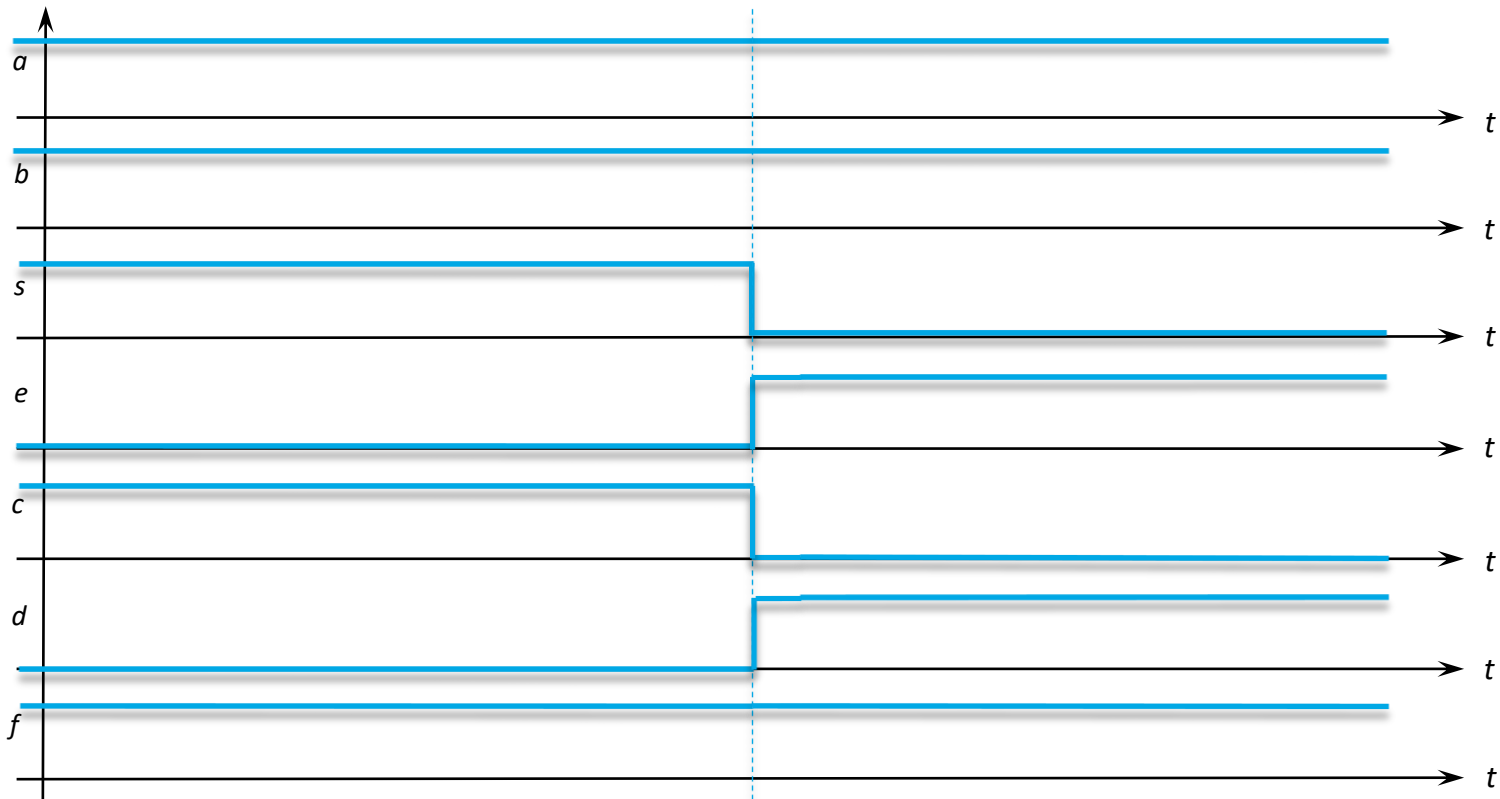
- ▶ **Sono brevi cambiamenti di valore che si esauriscono con il transitorio**
  - ▶ Sono dovuti a particolari combinazioni di ritardi nelle reti logiche
  - ▶ Influiscono solo sul transitorio
  - ▶ Possono dare molto fastidio quando si realizzano circuiti sensibili ai fronti dei segnali (circuiti sequenziali asincroni)

# Alee

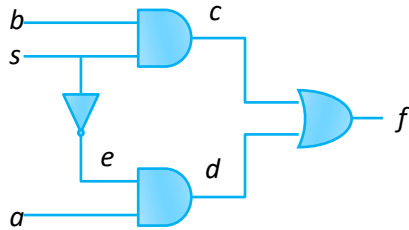


## ► In assenza di ritardi

- Un implicante si disattiva, l'altro si attiva
- L'uscita rimane al valore 1 costante

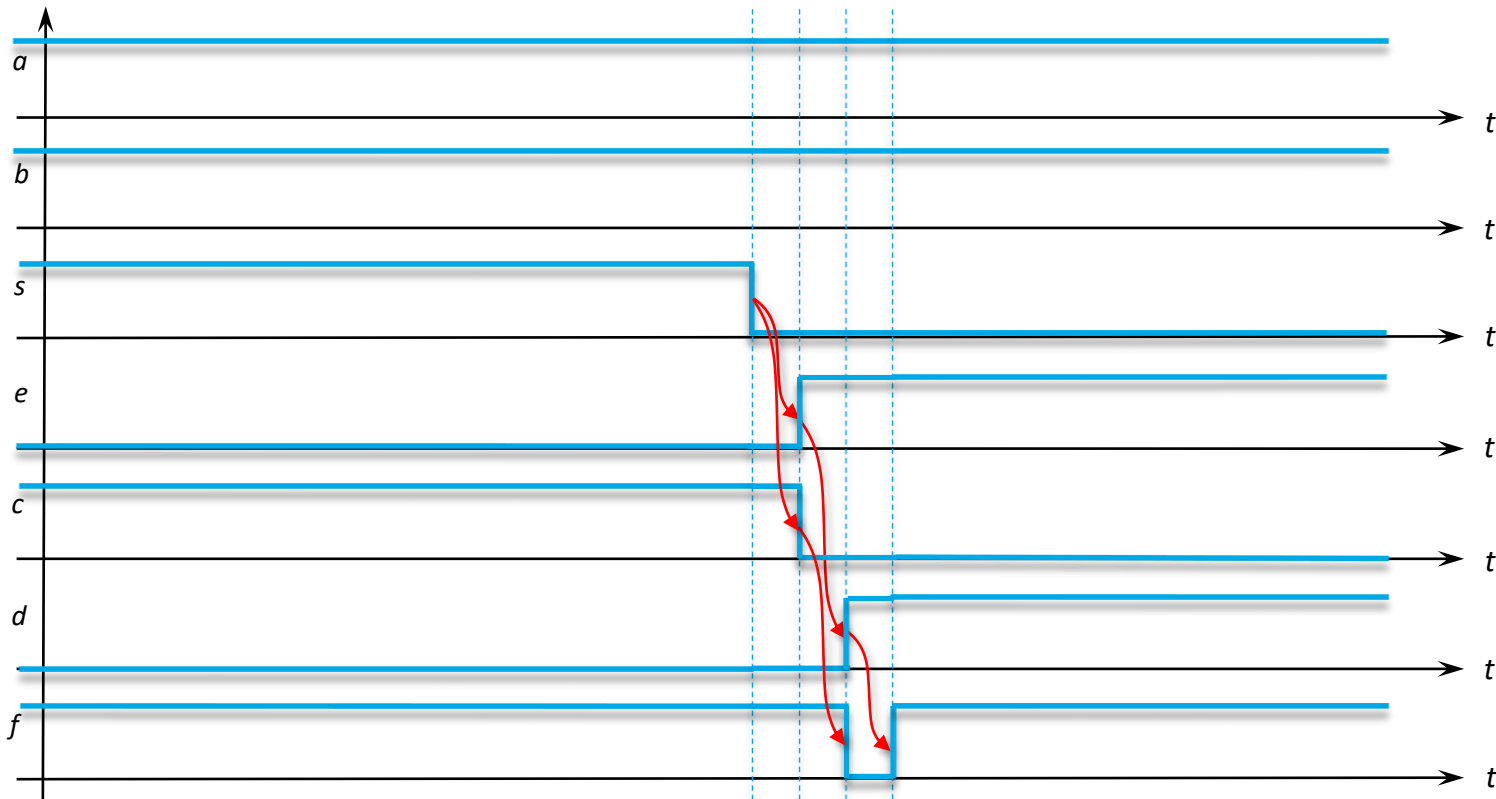


# Alee nel multiplexer



## ► Con ritardi

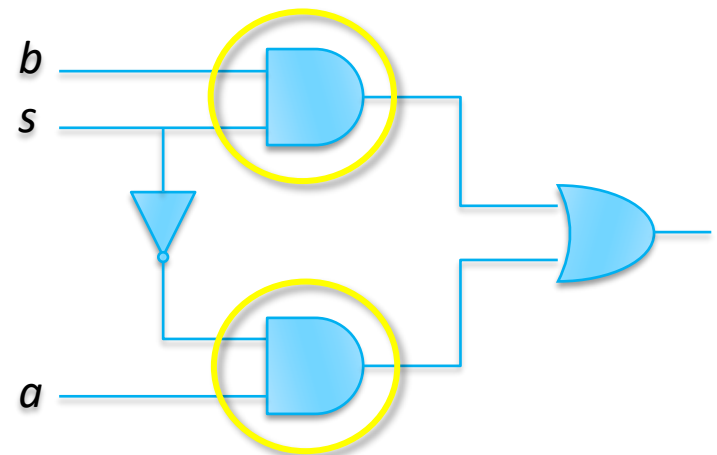
- L'implicante  $c$  si disattiva prima che l'implicante  $d$  possa mantenere l'uscita a livello alto
- L'uscita si porta per un periodo a livello basso



# Alee sulla mappa di Karnaugh

- ▶ Si verifica una alea solo se gli ingressi cambiano in modo da **passare da un implicante all'altro**
  - ▶ Inoltre i ritardi devono essere tali da **disattivare il primo, prima di attivare il secondo**
  - ▶ Nella mappa di Karnaugh si salta da una casella ad un'altra senza stare in uno stesso implicante
  - ▶ Se invece non si cambia implicante non si può verificare una alea

$s \backslash ab$	00	01	11	10
0	0	0	1	1
1	0	1	1	0

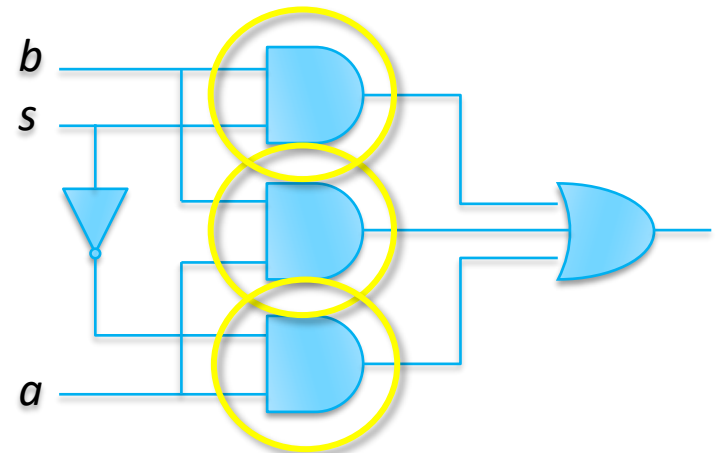


# Come prevenire le alee

## ► E' sufficiente aggiungere implicanti di collegamento

- La realizzazione non è più minima
- Indispensabile per i circuiti asincroni
- Utile nei circuiti sincroni solo per ridurre eventualmente i consumi
- Provate a costruire il diagramma temporale del circuito

$s \backslash ab$	00	01	11	10
0	0	0	1	1
1	0	1	1	0



- ▶ **Le tempistiche influenzano il funzionamento del circuito**
  - ▶ I ritardi introducono asimmetrie nella rete logica
  - ▶ Diversi modelli, con differente accuratezza, ci aiutano a stimare le tempistiche
  - ▶ Utile conoscere i ritardi minimi e massimi attraverso una rete logica
- ▶ **Asimmetrie danno luogo a fenomeni di alee**
  - ▶ Le alee, o glitch, possono aumentare i consumi di potenza
  - ▶ Sono particolarmente insidiose in certi tipi di circuiti sensibili ai fronti dei segnali
  - ▶ Si possono eliminare introducendo elementi che neutralizzano le transizioni spurie