

Bascule D

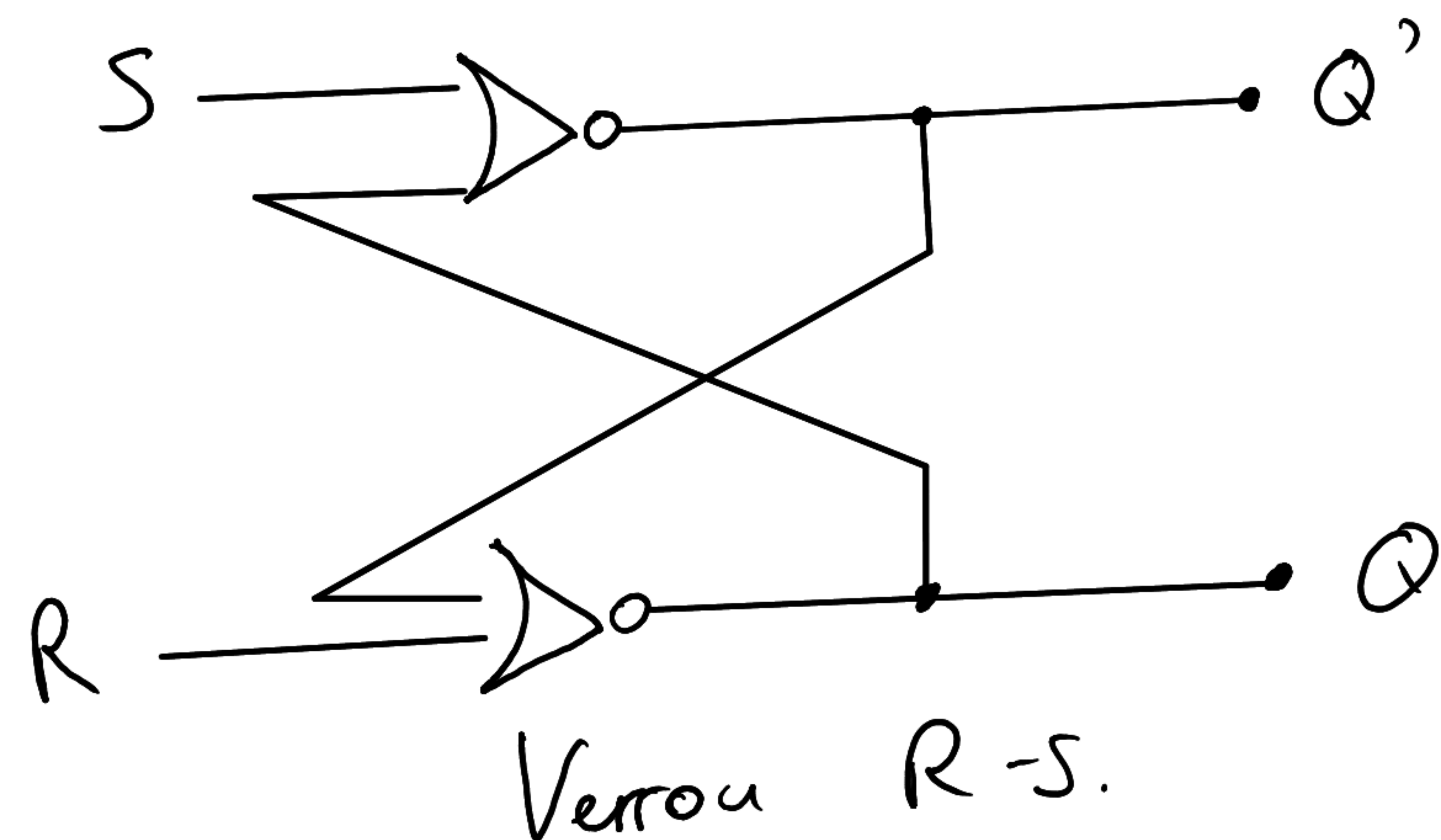
Motivation : Comprendre le fonctionnement de l'un des mécanismes de stockage d'un bit.

Inspiration : Design of digital computers, Gschwind et McCluskey

Bascule : Circuit logique capable de maintenir une valeur de sortie malgré des changements dans les valeurs d'entrées. $\rightarrow$ État "mémoire".

Bascule Verrou : Asynchrone ou sensible à un niveau d'horloge $\rightarrow$ Verrou
Synchrone et sensible à un front d'horloge $\rightarrow$ Bascule.

On va arriver à la bascule D en plusieurs étapes.



a	b	$a\bar{a}b$
0	0	1
0	1	1
1	0	1
1	1	0

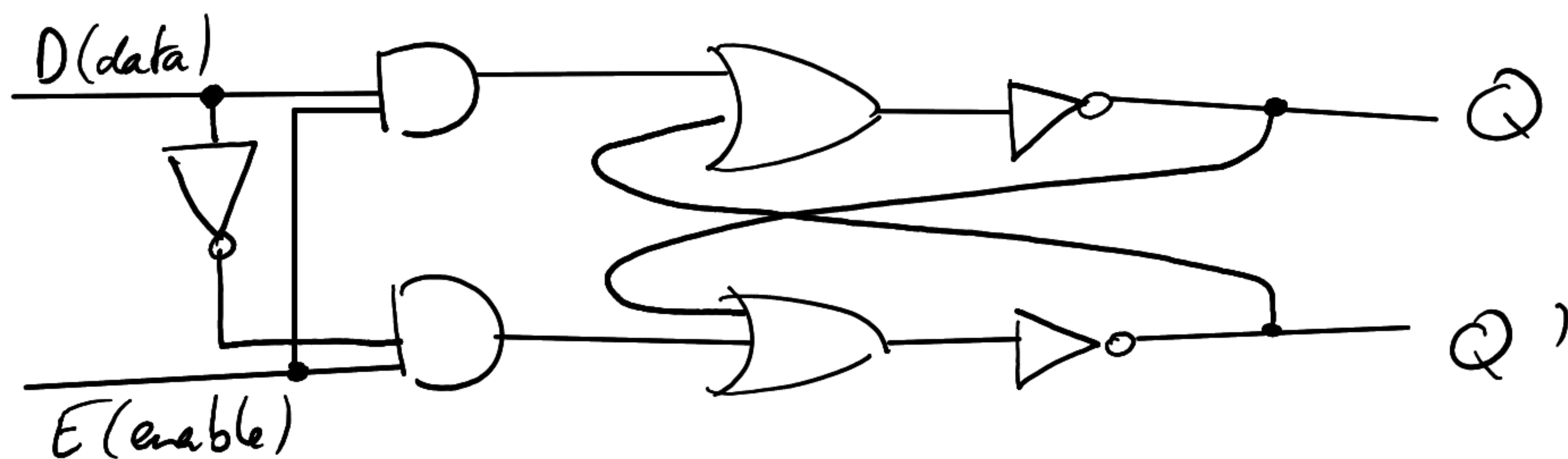
R : reset et S : set

R	S	Q	Q'
0	0	Q^-	Q'^- : valeurs inchangées
0	1	1	0
1	0	0	1
1	1	/	/

: état interdit.
car instable

En restant dans les états autorisés, on remarque :
 $\overline{Q} = Q'$.

On peut éviter un état interdit avec l'ajout de 3 portes logiques. On a alors le verrou D :

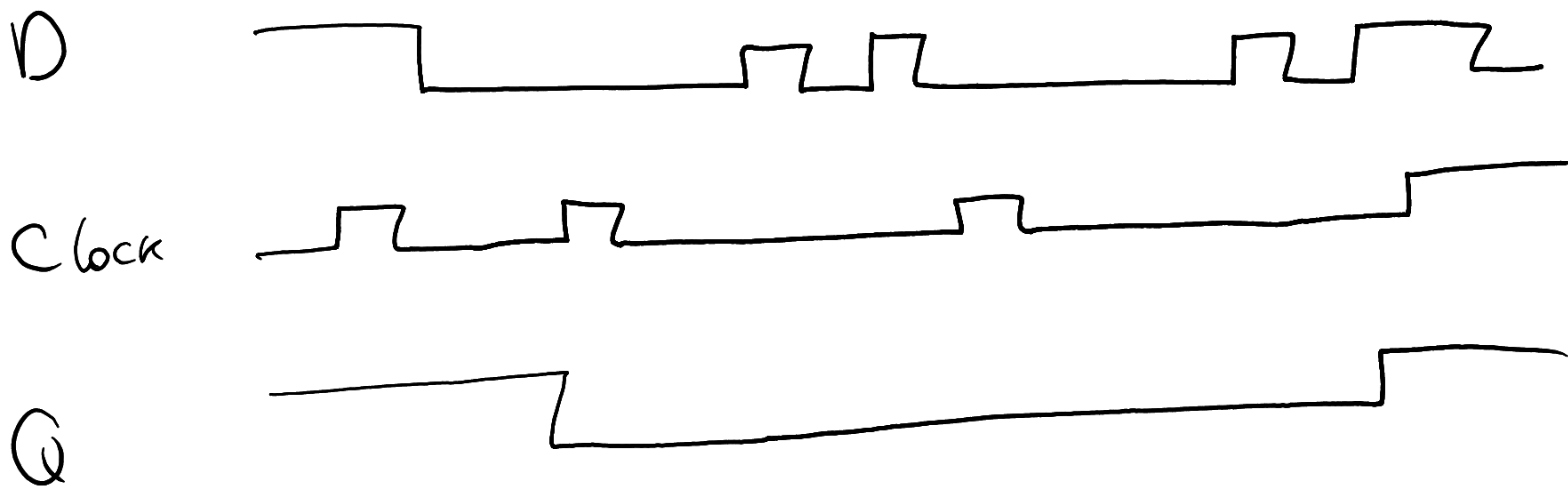
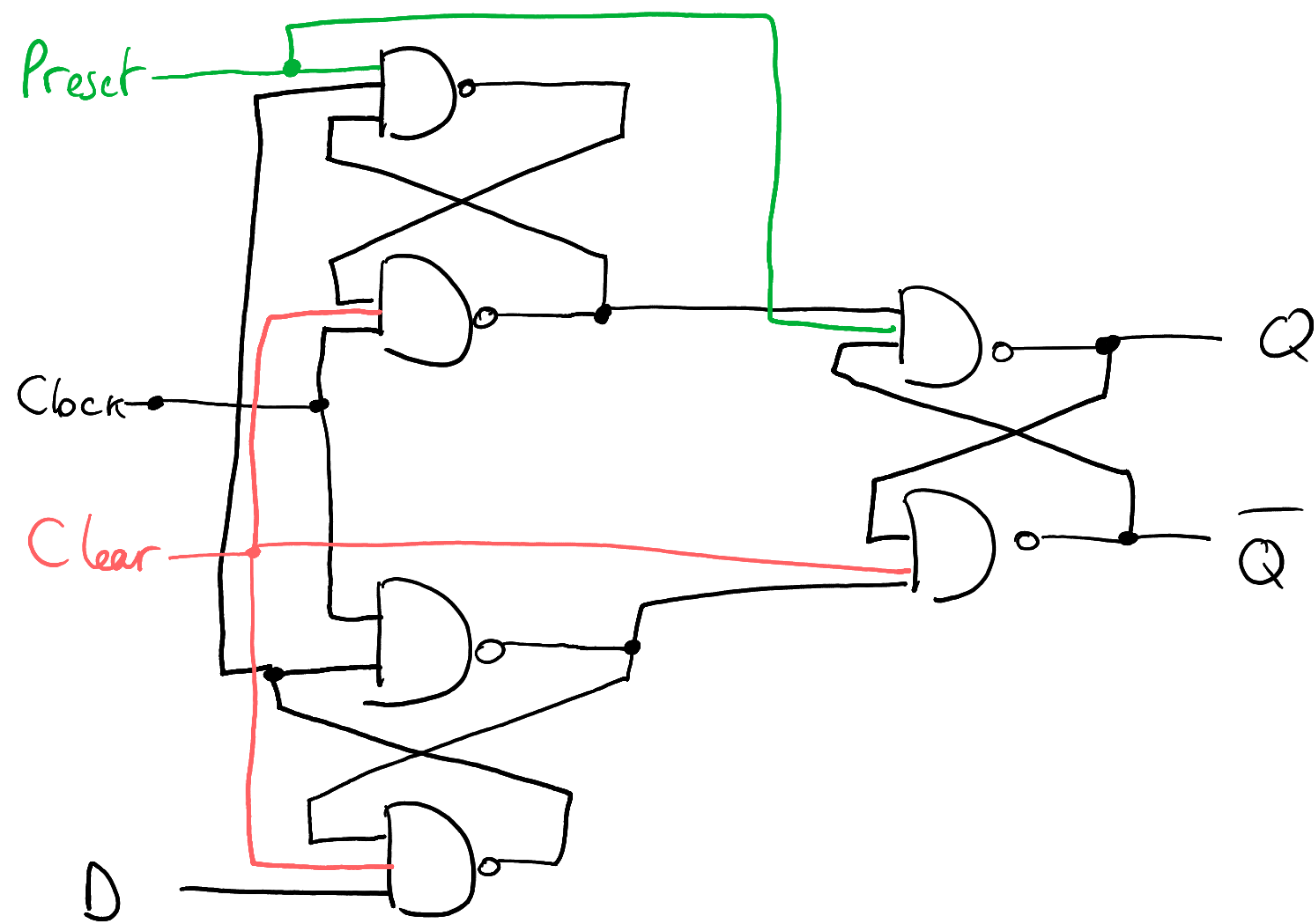


D	E	Q	
0	0	Q^-	} Valeur inchangée
1	0	Q^-	
0	1	0	
1	1	1	

- Si $E=0$, la valeur ne change pas.
- Si $E=1$, la valeur change pour celle de D.

Problème : Il y a beaucoup de situations où il y a besoin de connaître l'état d'une bascule et à la fois de changer son état (ex: compteurs, shifts, ...).

↳ Plusieurs façons de régler le pb: on présente ici la bascule D 5474/7474.



A' la différence du verrou D, la bascule D change d'état au front d'horloge et non a' la valeur.

Pour assurer le bon fonctionnement d'une bascule D, il y a plusieurs contraintes à respecter :

- le set-up time : le signal data ne doit pas bouger pdt ce temps ($\sim 15\text{ ns}$) : avant la clk

Clock

Data

$t_{\text{set-up}1}$

Data

$t_{\text{set-up}2}$

- le hold time : le signal data doit être stable pendant ce temps (2 ns) : après la clk.

Clock

Data

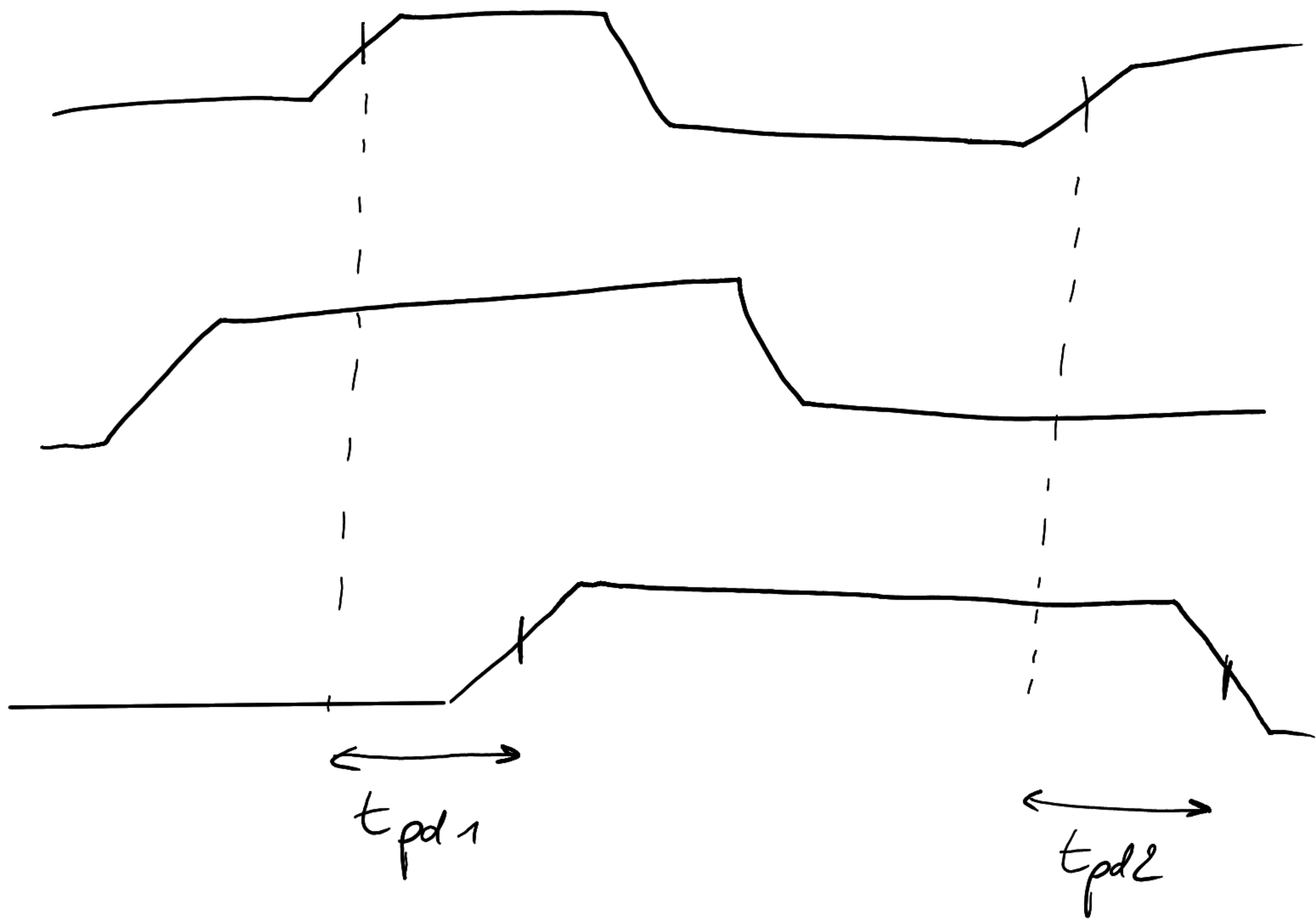
t_{hold}

Enfin, il faut tenir compte du temps de propagation:

Clock

Data

Q



$t_{pd1} \approx 16 \text{ ns}$ et $t_{pd2} \approx 22 \text{ ns}$.