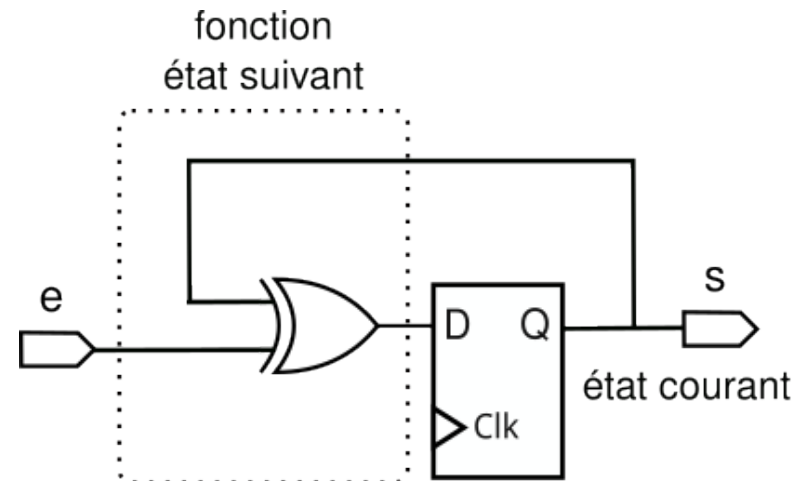


3– Automates ou *Machines d'états Finis* (FSM)

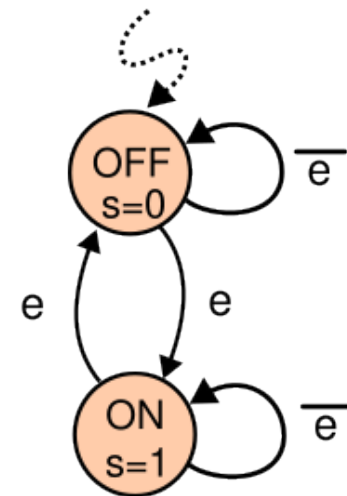
Du circuit au diagramme états-transitions

- **Rappel** : les valeurs des sorties Q représentent l'état du système à un instant logique donné.
- Le nombre d'états possibles est donc **fini**.
- Il est plus intuitif de les représenter *autrement* :

► Diagramme états transitions



e	Q	D	S
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

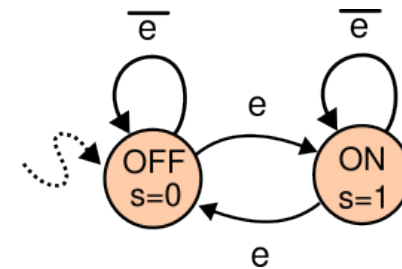


Note : nommage symbolique des états

Définition

Un diagramme états-transitions est un **graphe orienté valué** $G = (Q, A)$ où :

- Q : ensemble **fini** de sommets (les *états*).
- $E = \{e_1, \dots, e_n\}$: signaux d'entrée binaires ;
 $\Sigma = \{0, 1\}^n$: espace des combinaisons d'entrée.
- $S = \{s_1, \dots, s_m\}$: signaux de sortie binaires ;
 $\Lambda = \{0, 1\}^m$: espace des combinaisons de sortie.
- $\mathcal{C} \subseteq \mathcal{P}(\Sigma)$: ensemble de *conditions* (prédicats booléens sur les entrées).
- $A \subseteq Q \times \mathcal{C} \times \Lambda \times Q$: ensemble des *arcs orientés*.

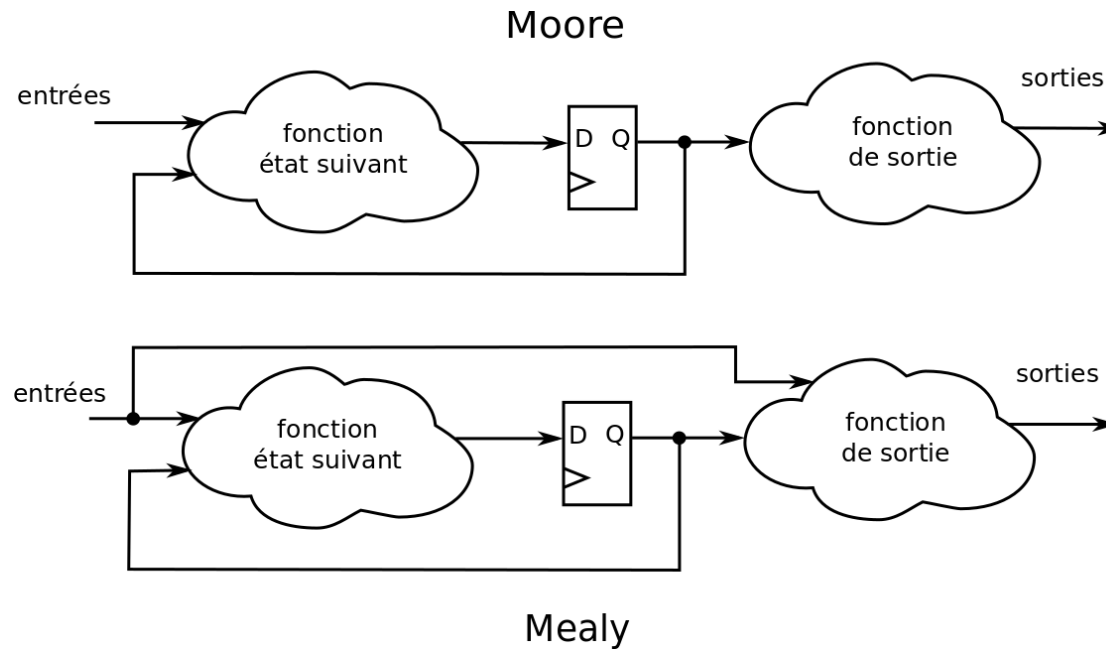


Sémantique d'un arc :

$$q_i \xrightarrow{c / \lambda} q_j \iff (q_i, c, \lambda, q_j) \in A \quad \text{avec } c \in \mathcal{C}$$

- $c \in \mathcal{C}$ est un **prédicat** sur Σ (ex : $e_1 \wedge \neg e_2$), permettant *don't-cares* et regroupement de combinaisons.
- **Moore** : λ dépend uniquement de q_i (sortie annotée sur le sommet).
- **Mealy** : λ dépend de q_i et de la combinaison d'entrée effective (sortie annotée sur l'arc).

Automates de Moore et Mealy : différences électroniques



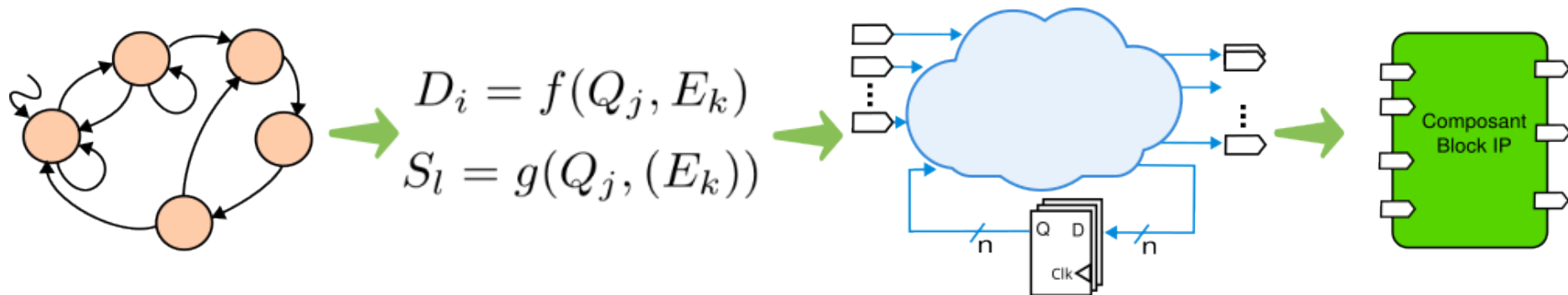
Warnings

- Composer (i.e connecter) des automates de Mealy est **dangereux**.
- Création potentielle de **boucles combinatoires** : **interdit**

Concevoir un circuit : du diagramme au circuit

Un problème de conception d'automate peut se résumer de la manière suivante :

- 1 Formulation du problème → diagramme états-transitions.
- 2 Vérification de la *causalité* de l'automate décrit.
- 3 Encodage des états.
- 4 Construction du tableau de séquençement (si nécessaire)
- 5 Détermination des fonctions d'état suivant (fonction de transition) et sorties.



Pour qu'un automate soit considéré comme consistant ou *causal*, on doit examiner les conditions $c_i(s)$ des transitions partant de chaque état $s \in Q$.

- **Réactivité** Les conditions sur les transitions doivent être *collectivement exhaustives* : il existe forcément une transition à '1'.

$$\forall s \in Q : \sum_i c_i(s) = 1$$

- **Déterminisme** Les conditions sur les transitions doivent être *mutuellement exclusives* : on ne peut pas avoir deux transitions à '1' en même temps.

$$\forall s \in Q, \forall (i, j), i \neq j : c_i(s).c_j(s) = 0$$

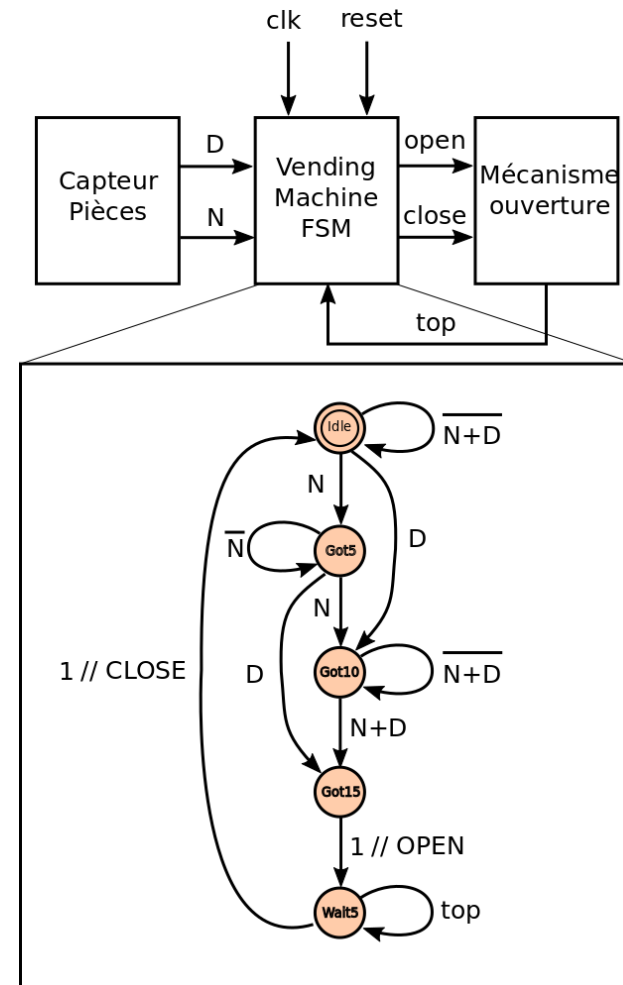
Retenir

Cela signifie simplement que dans un état, il existe un, et un seul, état suivant.

Notons qu'un état peut avoir comme état suivant lui-même : dans ce cas, le système décrit ne change pas d'état, tout simplement. En terme de diagramme états-transitions, cela revient à dessiner une boucle sur l'état (la "bulle").

Etude de cas : vending machine (distributeur)

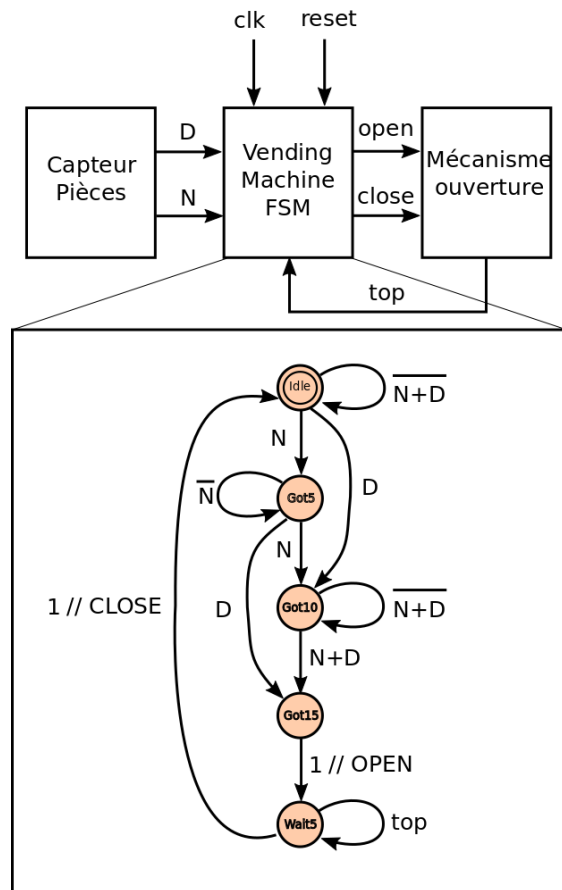
Objectif On cherche à réaliser un dispositif qui délivre un objet (canette etc), après que l'on y a introduit 15 centimes. Ces 15 centimes peuvent être introduits à l'aide de pièces de 5 (N) ou 10 centimes (D). La machine ne rend pas la monnaie.



L'encodage des états consiste à associer un code binaire à chaque nom symbolique des états.

- Une infinité d'encodages sont possibles.
- Chaque encodage conduit évidemment à des circuits différents structurellement, mais leur comportement aux entrées-sorties est identique.
- On retiendra **deux encodages particuliers** :
 - ① Encodage **dense** : numérotation binaire classique. Pour n états $\rightarrow \lceil \log_2(n) \rceil$ bits.
 - ② Encodage **"one-hot"** : **1 bit par état**. 1, 2, 4, 8, 16, 32, ... n états $\rightarrow n$ bits.
- L'encodage "one-hot" permet d'éviter les tableaux de Karnaugh.

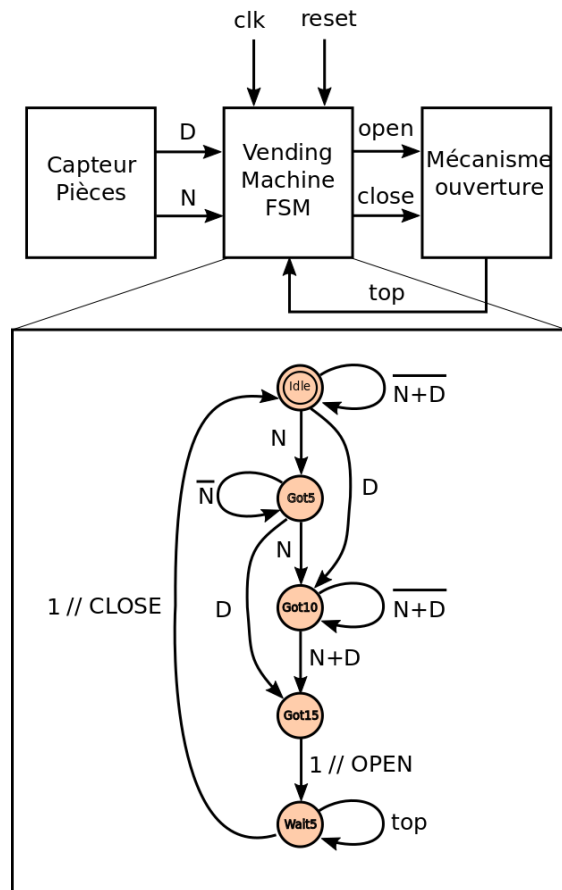
Etude de cas : Encodage dense



Encodage dense :

état	numéro	code binaire
Idle	0	000
Got5	1	001
Got10	2	010
Got15	3	011
Wait5	4	100

Etude de cas : encodage "one-hot"



Encodage "one-hot" :

état	numéro	code binaire
Idle	1	00001
Got5	2	00010
Got10	4	00100
Got15	8	01000
Wait5	16	10000

Etude de cas : table de séquençement

Le but est de trouver les équations :

$$D_i = f(Q_j, E_k)$$

$$S_l = g(Q_j, (E_k))$$

Dans un premier temps on conserve les noms des états symboliques.

état courant	e_1	...	e_n	état suivant	s_1	...	s_n
IDLE							
...							
GOT5							
...							

Puis on utilise l'encodage explicitant les Q_i (état courant) et D_i (état futur).

Q_n	...	Q_0	e_1	...	e_n	D_1	...	D_0	s_1	...	s_n

Equations finales : Il reste à observer les colonnes des D_i et s_j et déduire les équations logiques

- Déduction immédiate dans certains cas.
- Construction des tableaux de Karnaugh pour chaque D_i et s_j

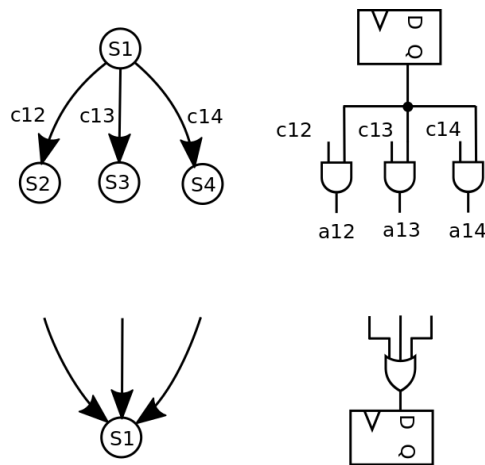
Cas Merveilleux de l'encodage One-Hot

On peut déduire les équations **directement** par simple **lecture** du diagramme états-transitions.

Principe

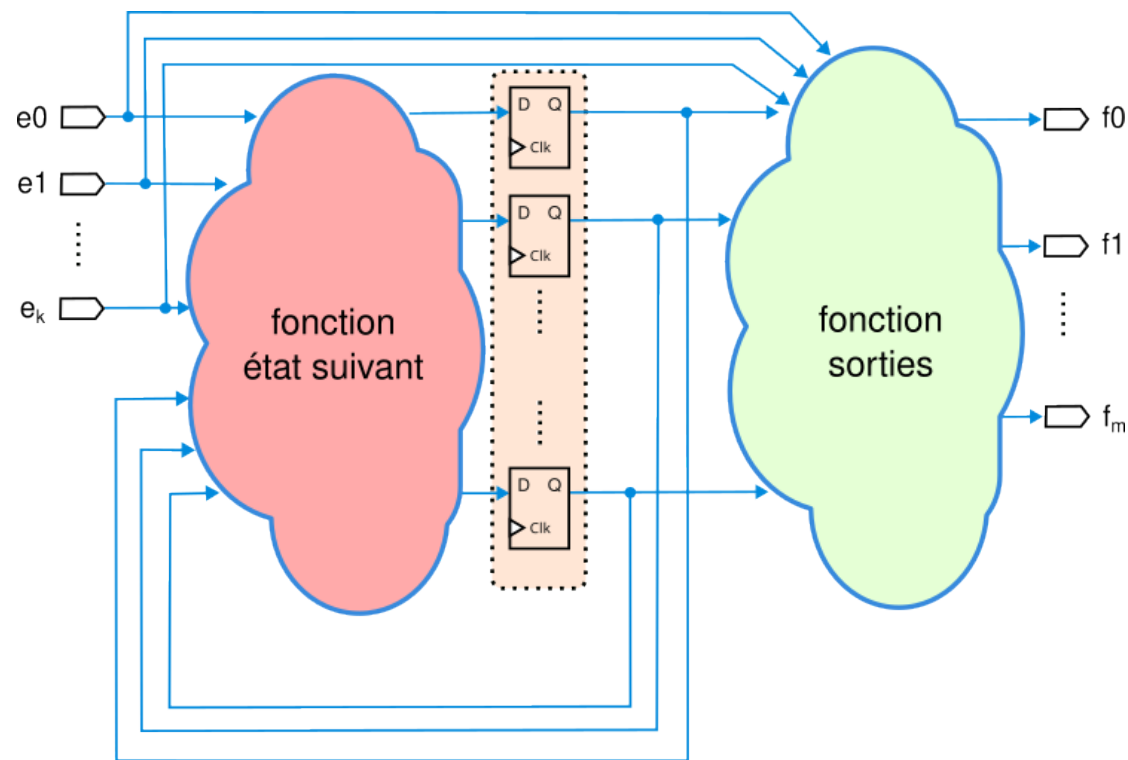
- Cet encodage spécifique signifie qu'une **seule** bascule D_i présente une sortie Q_i à 1.
- Tout se passe donc comme si un **jeton** circule d'une **bascule à l'autre**.
- Savoir quelle bascule "détient le jeton" suffit à connaître l'état courant.

Application directe du schéma de traduction suivant :



$$\begin{cases} d_2 = q_1 \cdot c_{12} + \dots \\ d_3 = q_1 \cdot c_{13} + \dots \\ d_4 = q_1 \cdot c_{14} + \dots \\ d_1 = \dots + \dots + \dots \end{cases}$$

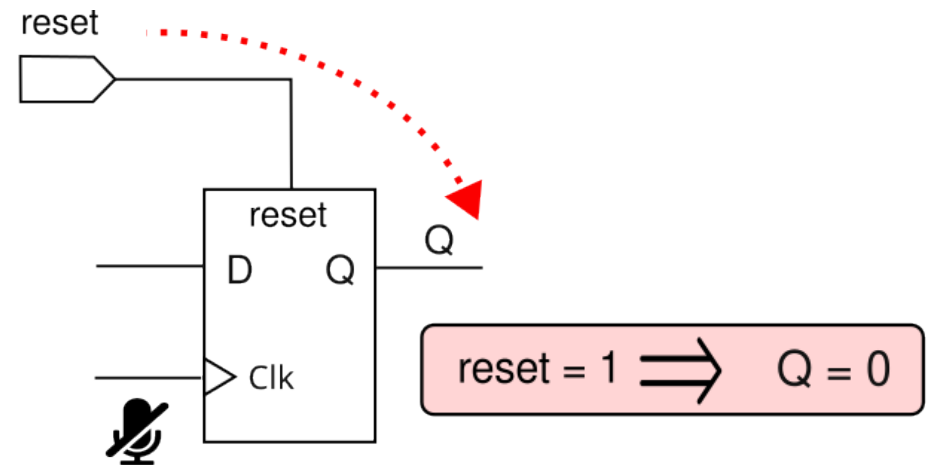
Dans certains cas simples, il reste possible de dessiner le circuit. Sinon, on s'en tient au schéma d'ensemble suivant :



Bascule D : Reset et Set *asynchrones*

Cas du Reset : mise à 0

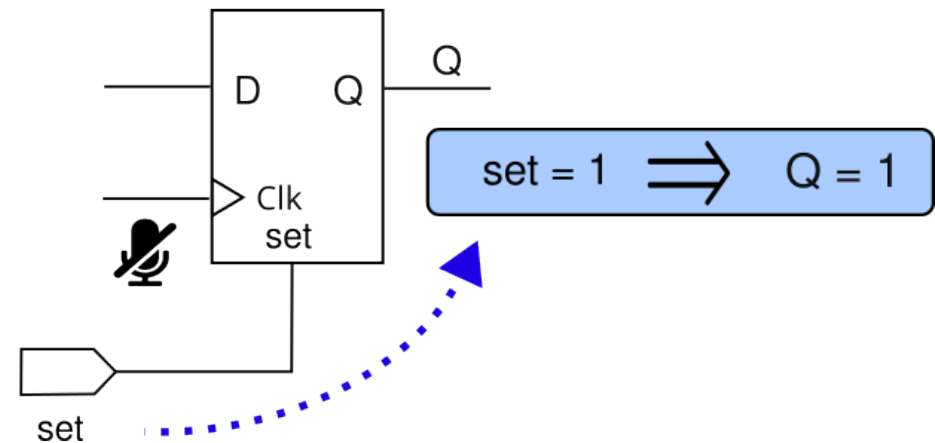
- Le reset est **prioritaire** sur l'horloge.
- Lorsqu'il est activé $\rightarrow Q = 0$



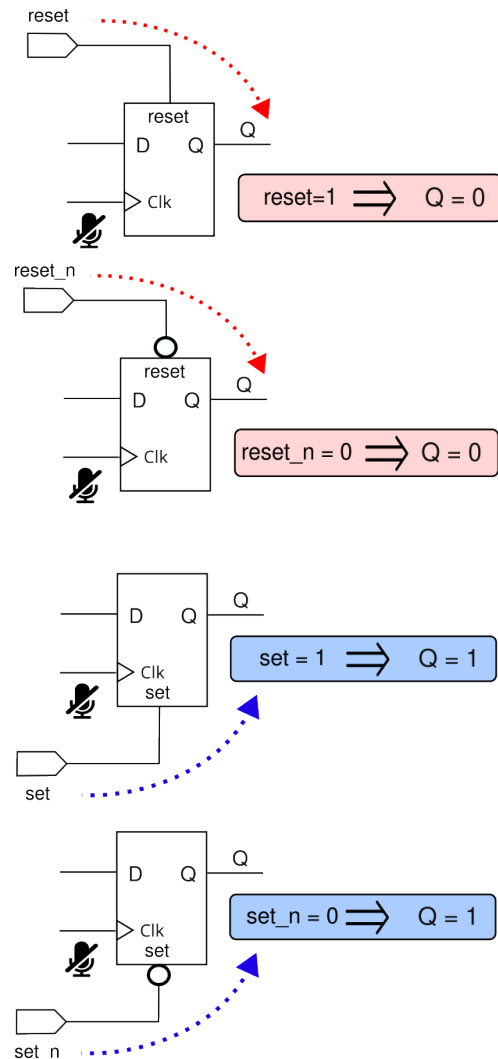
Cas du Set : mise à 1

- Le set est **prioritaire** sur l'horloge.
- Lorsqu'il est activé $\rightarrow Q = 1$

En cas d'ordres contradictoires $\rightarrow$ voire doc constructeur.



Bascule D : Reset et Set *asynchrones*



Polarité d'activation

- Les deux signaux reset et set ont eux-mêmes une valeur où ils sont actifs.
- Cette valeur est précisée dans le schéma de la bascule :
 - ▶ Cerle ° indique la polarité 0 active.
 - ▶ L'absence de cerle ° indique la polarité 1 active.
- Cette valeur *peut* être précisée dans le *nommage* du signal :
 - ▶ reset_n indique une polarité 0 active.
 - ▶ reset indique une polarité 1 active.

Ceci dépend des *conventions de nommage* propres aux entreprises.