



HAL
open science

VHDL Avancé 3 : Métastabilité

Clément Foucher

► **To cite this version:**

Clément Foucher. VHDL Avancé 3 : Métastabilité. Licence. VHDL Avancé, Toulouse, France. 2025.
hal-04955310

HAL Id: hal-04955310

<https://hal.science/hal-04955310v1>

Submitted on 18 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License

VHDL avancé

MÉTASTABILITÉ

© 2025 Clément Foucher

Cours distribué sous licence libre 

BUT GEII Toulouse S6 ESE

Circuits synchrones : théorie et réalité

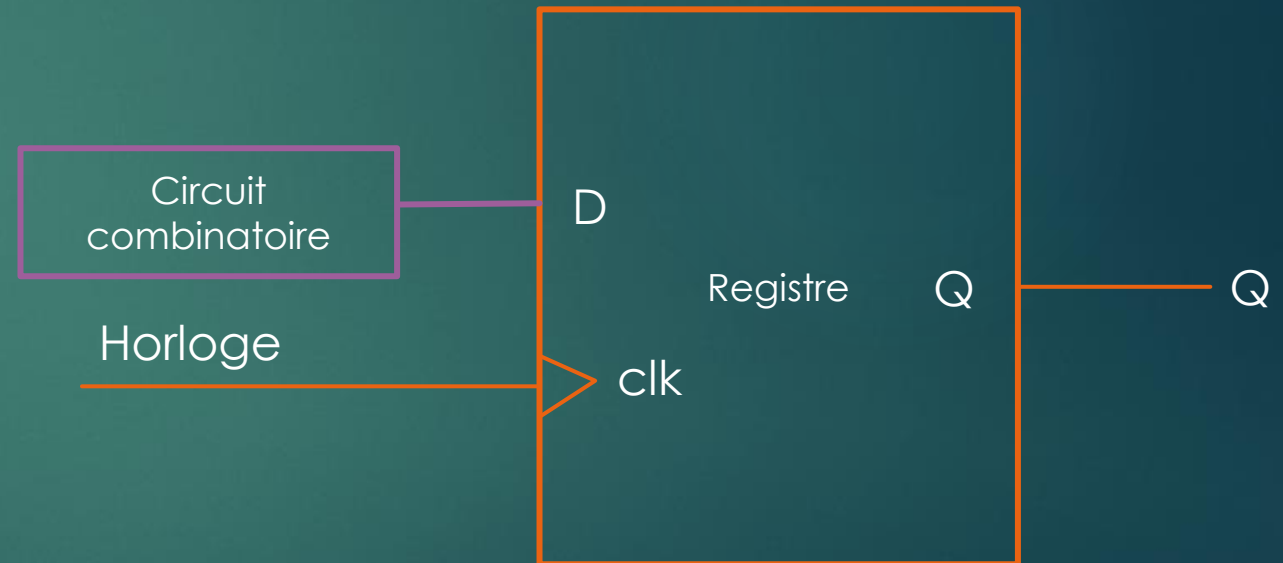
LA MÉTASTABILITÉ

MÉTA-STABILITÉ : CAUSES ET RISQUES

EN PRATIQUE EN VHDL

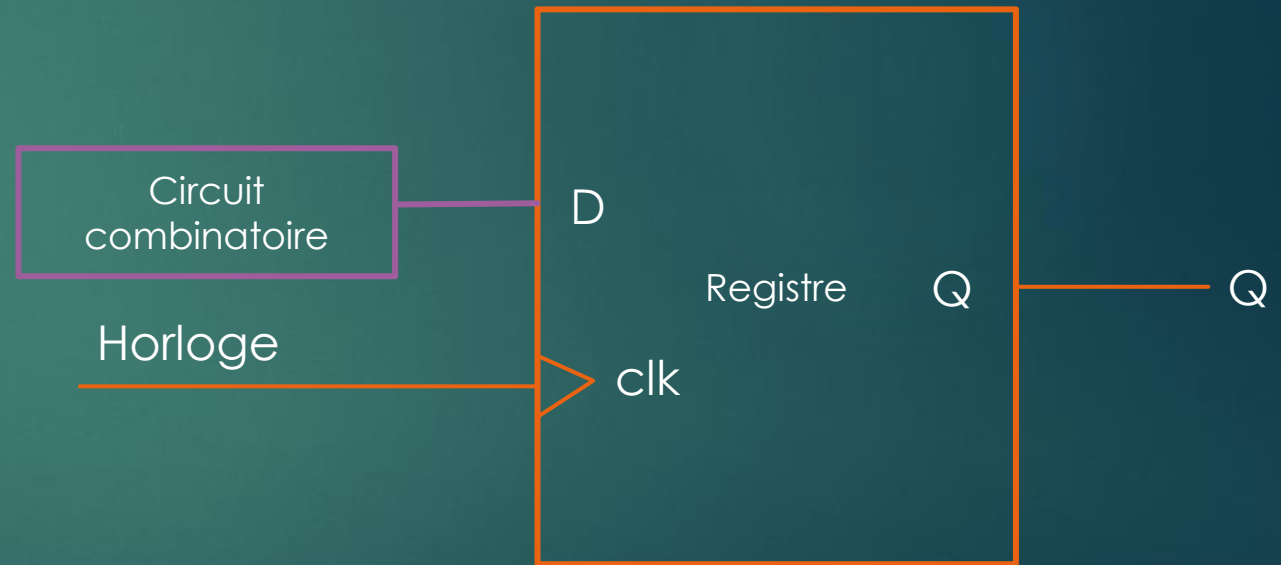
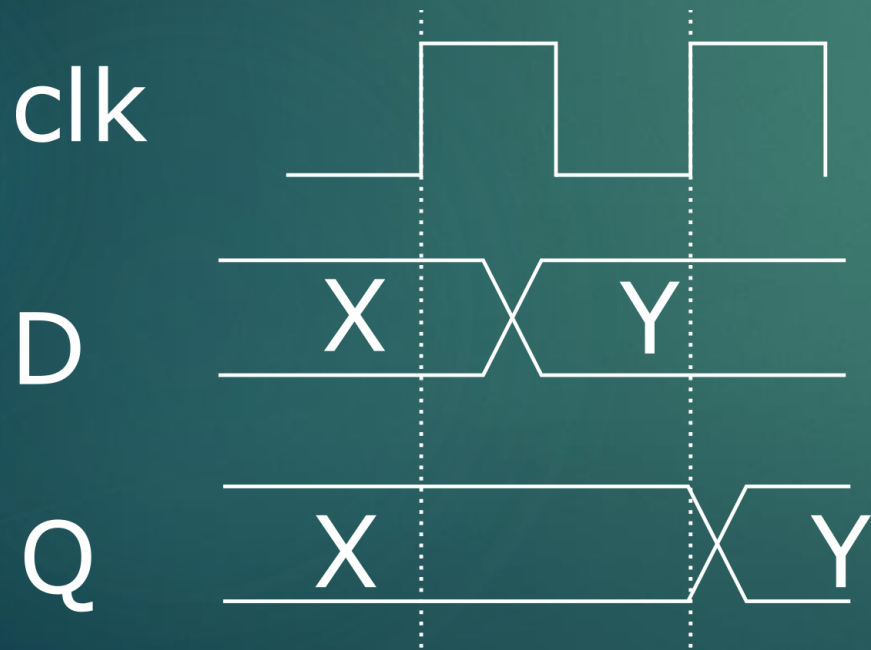
Circuit synchrone

- ▶ Un circuit synchrone se définit par une évolution *synchronisée* par une horloge
- ▶ Un circuit synchrone est constitué de logique combinatoire (asynchrone) et de bascules ou registres (synchrone)



La théorie

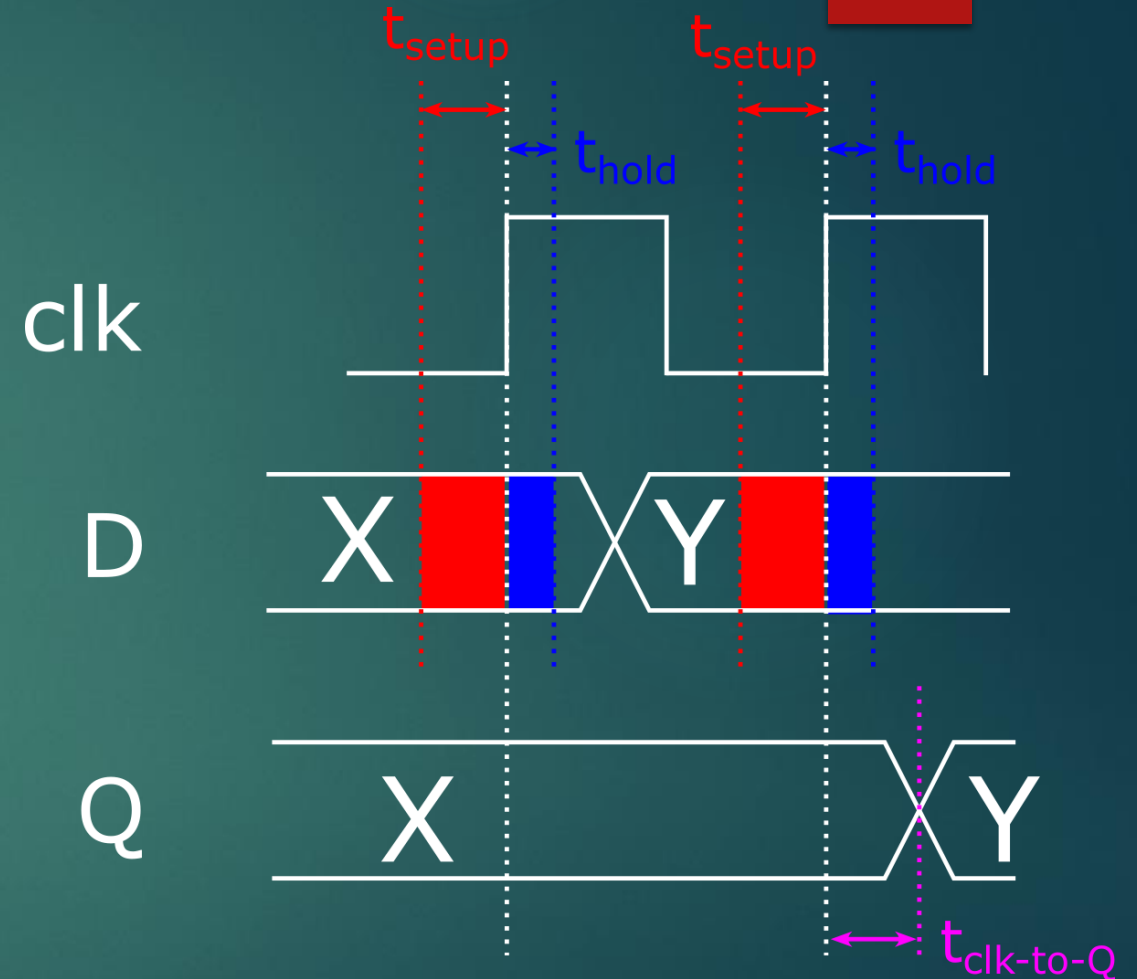
- ▶ Si D change de valeur entre deux fronts d'horloge, Q prendra la nouvelle valeur immédiatement après le front de l'horloge



La réalité

5

- ▶ Une bascule est constituée de transistors
- ▶ Les transistors ne changent pas d'état instantanément
 - ▶ Il y a des durées imposées par la physique
 - ▶ Ces durées nécessitent que les entrées de la bascule soient stable un certain temps pour que la mémorisation opère correctement
- ▶ Spécification d'une bascule
 - ▶ t_{setup} : temps pendant lequel l'entrée doit être stable avant le front
 - ▶ t_{hold} : temps pendant lequel l'entrée doit être stable après le front
 - ▶ $t_{\text{clk-to-Q}}$: temps après lequel la sortie est mise à jour après un front
- ▶ Que se passe-t-il si un des temps de stabilité nécessaire sur les entrées est violé ?



D ne doit pas changer de valeur dans la zone rouge ni la zone bleue

CIRCUITS SYNCHRONES : THÉORIE ET RÉALITÉ

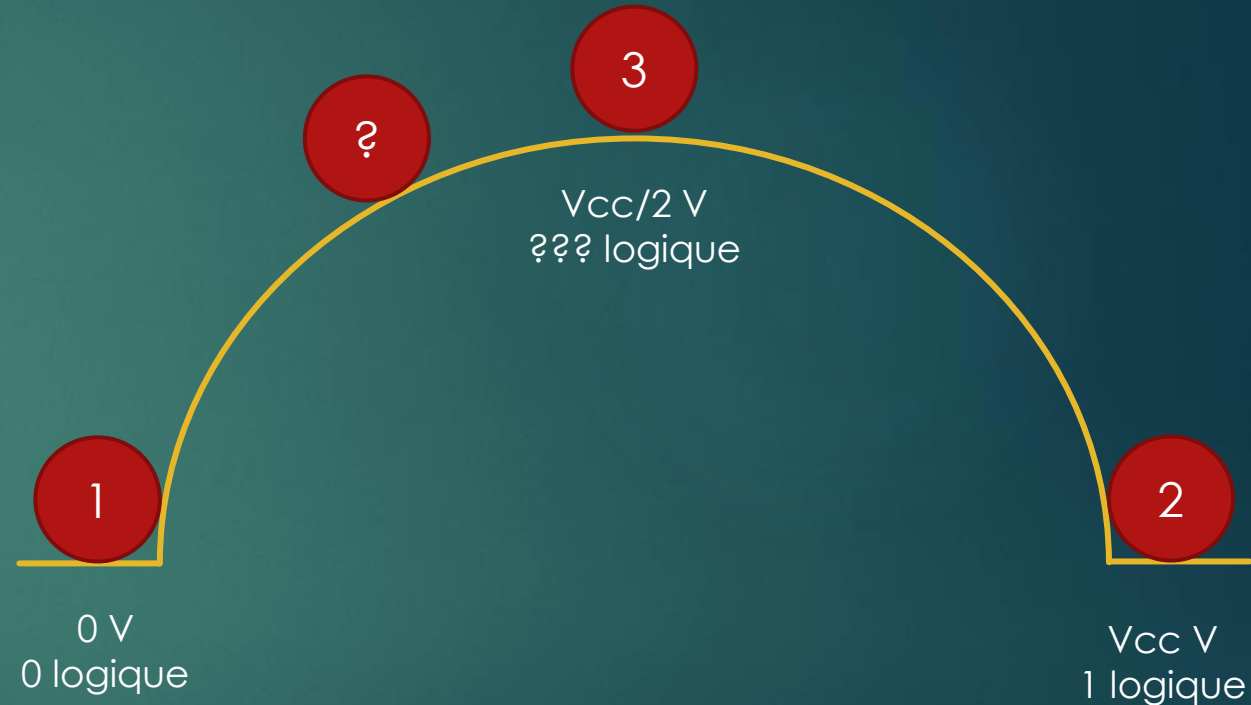
La métastabilité

MÉTA-STABILITÉ : CAUSES ET RISQUES

EN PRATIQUE EN VHDL

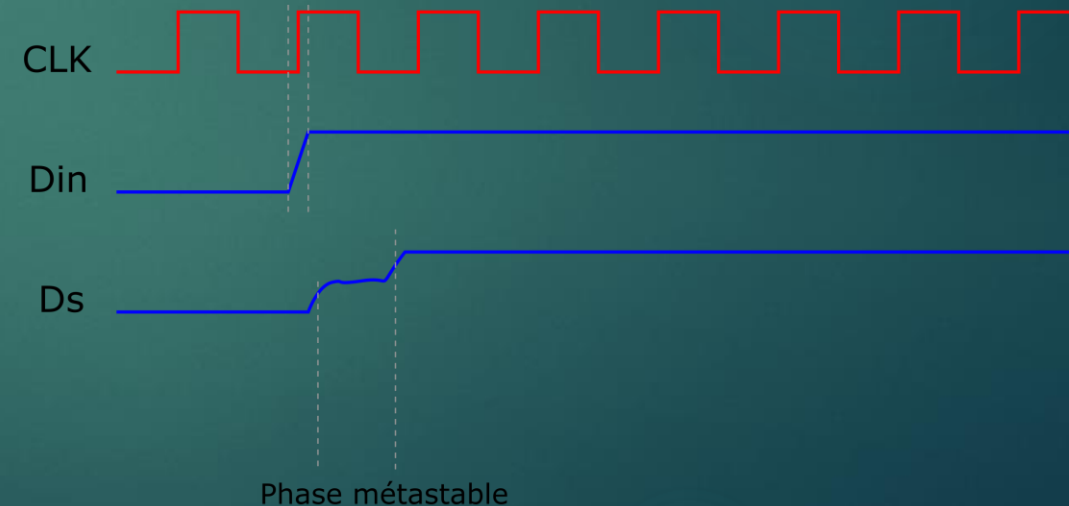
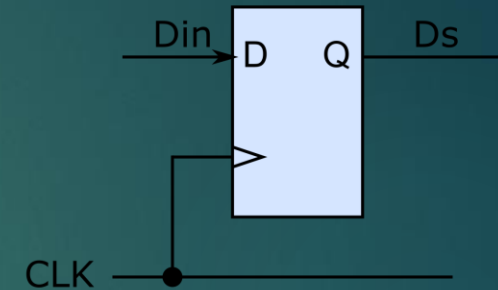
Une bascule

- ▶ Analogie avec la gravité
 - ▶ Soit une colline et une boule
 - ▶ Quels sont les points stables ?
- ▶ La boule 3 est stable... tant qu'on ne la pousse pas
- ▶ Les niveaux de tension d'une bascule fonctionnent d'une manière similaire
 - ▶ *On ne rentrera pas dans les détails technologiques du « pourquoi », mais c'est physiquement inévitable*
- ▶ Le viol d'un des temps de stabilité requis par la bascule peut avoir pour effet de laisser la tension « en haut de la colline »



La durée de méta-stabilité

- ▶ La méta-stabilité se résout normalement d'elle-même
 - ▶ En effet, elle n'est pas... stable
 - ▶ Toute perturbation (thermique, électromagnétique, etc.) contribuera à faire « bouger la boule »
- ▶ Toute la question est donc de savoir combien de temps elle met à se résoudre
 - ▶ La probabilité de rester dans un état méta-stable décroît de manière exponentielle avec le temps



CIRCUITS SYNCHRONES : THÉORIE ET RÉALITÉ
LA MÉTASTABILITÉ

Méta-stabilité : causes et risques
EN PRATIQUE EN VHDL

Risques associés à la méta-stabilité

10

- ▶ Un signal méta-stable en sortie d'une bascule peut « contaminer » toutes les portes combinatoires auxquelles il est relié
- ▶ L'entrée des bascules qui utilisent ces signaux peut se retrouver à une valeur qui ne vaut... ni 0 ni 1
 - ▶ Elles entreront donc en méta-stabilité elles-mêmes au prochain cycle d'horloge
- ▶ L'ensemble de ces contaminations peut totalement planter le circuit numérique
- ▶ Il est donc nécessaire d'identifier les causes potentielles de méta-stabilité et les traiter

Causes de la méta-stabilité

- ▶ Plusieurs causes possibles
 - ▶ « durée » de calcul combinatoire trop longue
 - ▶ Changement de zone d'horloge
 - ▶ Signal en provenance du monde extérieur

Causes de la méta-stabilité :

« durée » de calcul combinatoire trop longue

- ▶ Le cumul du temps de parcours dans les portes logiques est trop long vis-à-vis de la période d'horloge
 - ▶ Le signal en sortie du circuit combinatoire n'est pas encore stabilisé au moment du front d'horloge
- ▶ Détection
 - ▶ Les logiciels de design pour FPGA sont capables de calculer le chemin critique (« critical path »), c'est-à-dire le chemin combinatoire le plus long
 - ▶ Ils affichent donc un warning ou une error si cette durée est trop longue vis-à-vis de la période
 - ▶ Résolution
 - ▶ Réduire le chemin critique (pas forcément aisé)
 - ▶ Diminuer la fréquence d'horloge (et donc augmenter sa période)

Causes de la méta-stabilité :

Changement de domaine d'horloge

- ▶ Dans un FPGA, il peut y avoir plusieurs horloges différentes, avec des fréquences et phases différentes
- ▶ Un signal qui est produit dans le domaine A (et est donc synchrone à l'horloge A) et utilisé dans le domaine B n'est pas forcément synchrone à l'horloge B
- ▶ Détection
 - ▶ Vous n'aurez pas ce cas en TP, nous n'utiliserons qu'une seule horloge
 - ▶ Les logiciels sont également capables de détecter un certain nombre de ces erreurs
- ▶ Résolution
 - ▶ Ajout d'une « zone tampon » entre les deux domaines, donc l'écriture est synchrone à la zone A et la lecture synchrone à la zone B

Causes de la méta-stabilité :

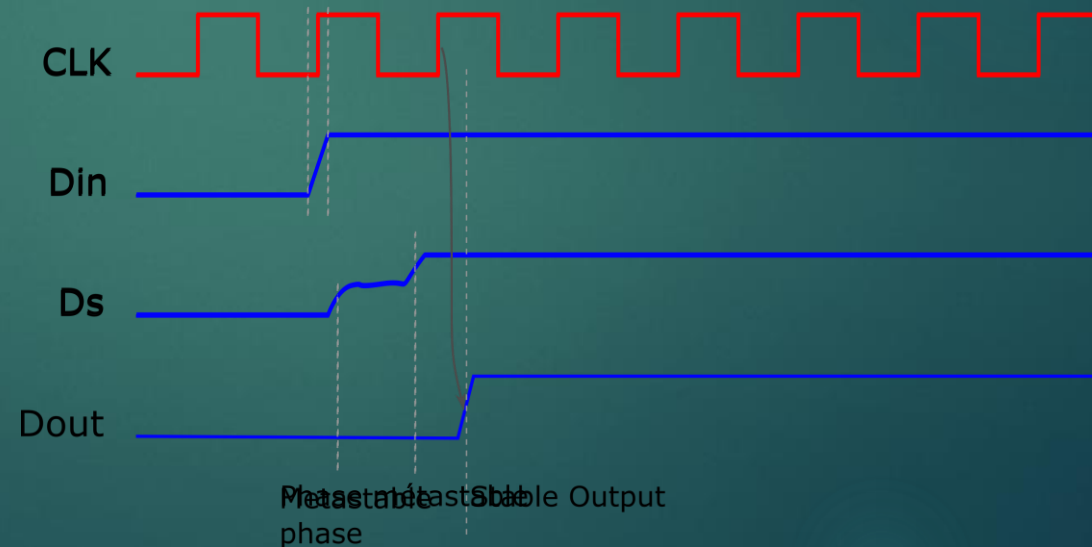
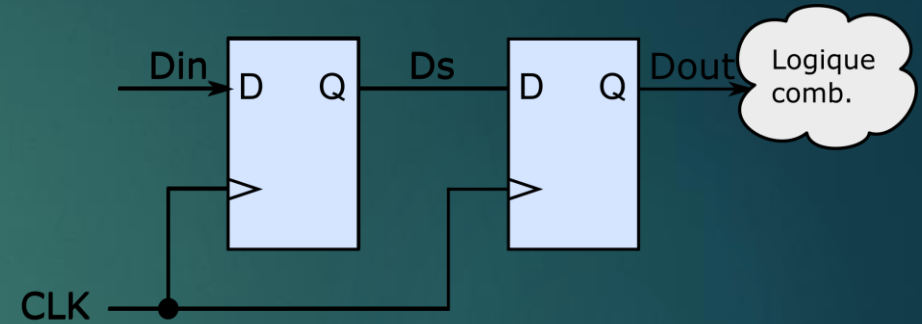
Signal en provenance du monde extérieur

- ▶ Un signal qui n'est pas produit par le circuit (exemple : patte d'entrée d'un FPGA) n'est clairement pas synchrone à son horloge interne
- ▶ C'est la plus grande cause de méta-stabilité
- ▶ C'est la seule dont on se préoccupera dans ce module
- ▶ Détection
 - ▶ Concerne *tous* les signaux d'entrée en provenance des GPIO du FPGA
 - ▶ Il faut donc systématiquement « immuniser » ces entrées contre la méta-stabilité
- ▶ Résolution
 - ▶ Chaîne de bascules D en série

Résoudre la méta-stabilité sur les entrées

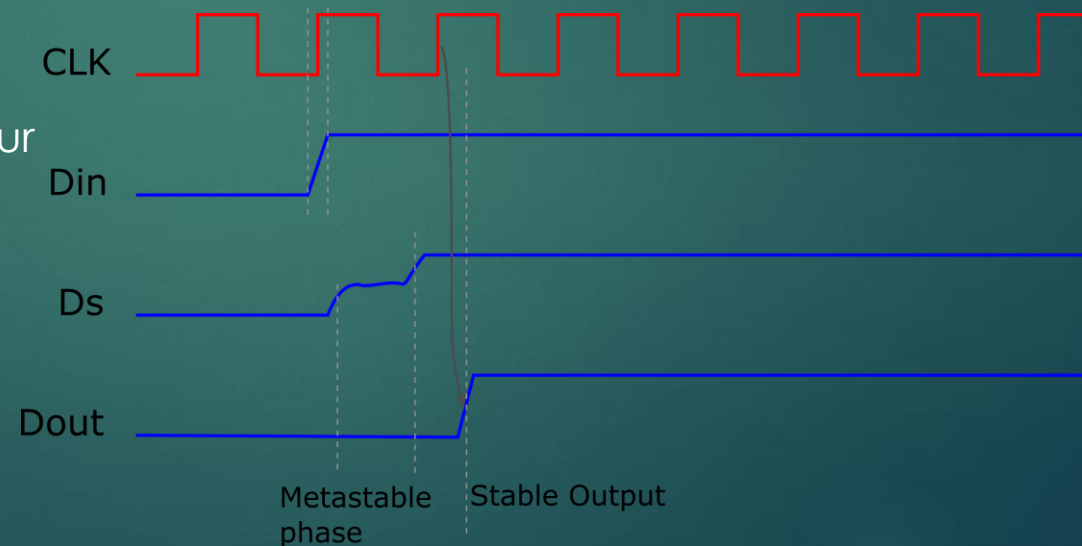
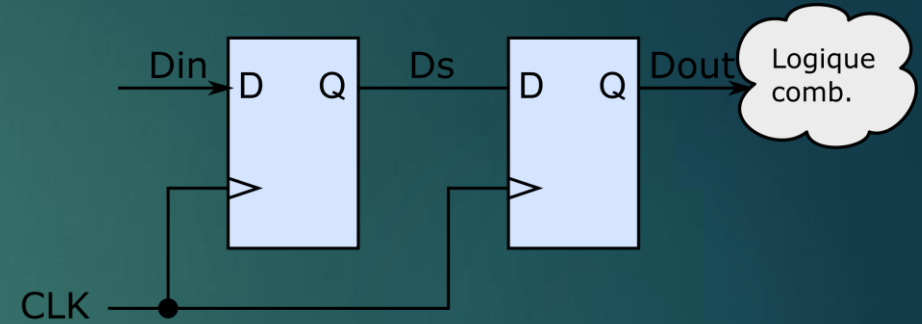
15

- ▶ Dans l'exemple précédent, si la durée de méta-stabilité n'excède pas un cycle...
- ▶ Il est possible de rajouter une seconde bascule en série avec la première pour neutraliser le problème



Résoudre la méta-stabilité sur les entrées

- ▶ Et si la durée est plus longue qu'un cycle ?
- ▶ Les probabilités se calculent et diminuent d'autant qu'on ajoute des bascules en série
- ▶ Dans la plupart des cas, deux bascules suffisent
 - ▶ Trois pour être ultra-certain pour des applications critiques

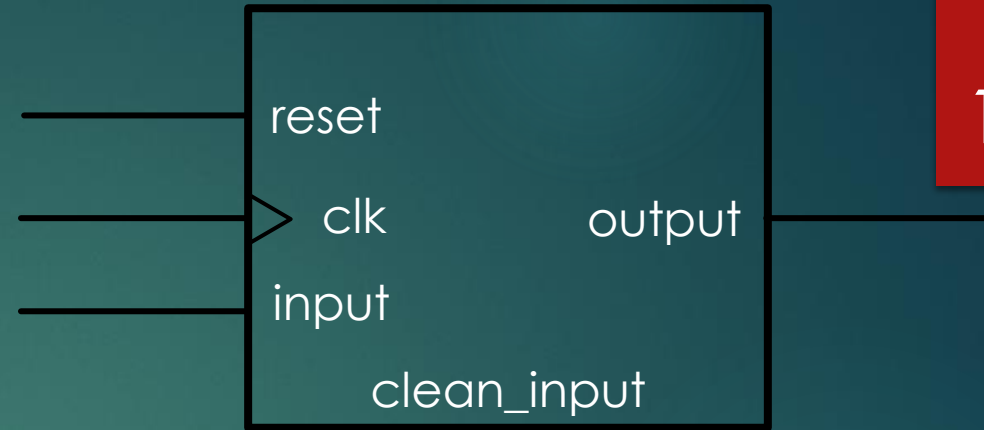


CIRCUITS SYNCHRONES : THÉORIE ET RÉALITÉ
LA MÉTASTABILITÉ
MÉTA-STABILITÉ : CAUSES ET RISQUES

En pratique en VHDL

En VHDL

- ▶ 2 bascules D : suffisant pour nous
- ▶ Créer un composant tout simple pour « nettoyer » les entrées
- ▶ On peut en profiter pour faire d'autres traitements si nécessaire
 - ▶ Anti-rebond par exemple



18

```
architecture ar of clean_input is
    signal i_signal : std_logic;
begin
    process(clk, reset)
    begin
        if reset='1' then
            i_signal <= '0';
            output <= '0';
        elsif rising_edge(clk) then
            i_signal <= input;
            output <= i_signal;
        end if;
    end process;
end architecture;
```