

---

# ***Electronique Numérique***

## ***2ème tome***

### ***Systemes séquentiels***

*Etienne Messerli  
Yves Meyer  
décembre 2004  
Version 1*

## Mise à jour de ce manuel

La base du présent manuel a été écrite par M. Yves Meyer de l'école d'ingénieurs de l'arc jurassien. J'ai repris celui-ci en apportant des modifications. La principale modification concerne la suppression du chapitre sur le langage VHDL. Celui-ci est couvert à l'EIVD par un manuel séparé.

Je remercie M. Yves Meyer de sa collaboration et de m'avoir permis de réutiliser son support de cours.

J'ai aussi repris des parties des manuels écrits par M. Maurice Gaumain. Il s'agit principalement des chapitres sur les multiplexeurs, les décodeurs et les aspects techniques.

Je remercie tous les utilisateurs de ce manuel de m'indiquer les erreurs qu'il comporte. De même, si des informations semblent manquer ou sont incomplètes, elles peuvent m'être transmises, cela permettra une mise à jour régulière de ce manuel.

## Contact

Auteur: Etienne Messerli  
e-mail : etienne.messerli@eivd.ch  
Tél: +41 (0)24 / 423.22.82

Adresse: Ecole d'Ingénieurs du Canton de Vaud (EIVD)  
Route de Cheseaux 1  
CH-1400 Yverdon-les-Bains  
Tél : ++41 (0)24 / 423.21.11 (central)  
Fax : ++41 (0)24/425.00.50  
E-mail : reds@eivd.ch  
Internet : <http://reds.eivd.ch>

## Autres personnes à contacter en cas d'absence:

M. Boada Serge	e-mail : Serge.Boada@eivd.ch	Tél. direct +41 (0)24/425.22.53
M. Bornand Cédric	e-mail Cedric.Bornand@eivd.ch	Tél. direct +41 (0)24/423.97.51
M. Sultan Ali	e-mail Ali.Sultan@eivd.ch	Tél. direct +41 (0)24/424.22.53

# Table des matières

<b>Chapitre 1 LES BASCULES</b>	<b>1</b>
1-1. Définition du système séquentiel .....	2
1-2. Bascule R-S en portes NAND .....	2
1-3. Bascule R-S en portes NOR .....	3
1-4. Bascule R-S avec enable .....	4
1-5. L'élément mémoire D (D Latch) .....	5
1-6. La bascule maître-esclave. ....	6
1-7. La bascule bistable déclenchable sur un flanc "edge-triggered" .....	8
1-8. Signal d'horloge et bascules synchrones .....	9
1-9. FLIP-FLOP D .....	10
1-10. BASCULE T (Toggle flip flop) .....	10
1-11. Entrées asynchrones .....	10
1-11.1. Désignation des entrées asynchrones .....	11
1-12. Considérations sur la synchronisation des bascules .....	11
1-12.1. Temps de stabilisation (setup time) et temps de maintien (hold time) .	12
1-12.2. Temps de propagation .....	12
1-13. Description d'une bascule D avec reset asynchrone en VHDL .....	13
1-14. Synchronisation des entrées externes asynchrones .....	14
<b>Chapitre 2 Synchronisation et détection des flanc d'un signal externe asynchrone</b>	<b>17</b>
<b>Chapitre 3 Les Registres</b>	<b>19</b>
3-1. Structures de base des registres .....	19
3-2. Registre Tampon (registre parallèle) .....	20
3-2.1. Description VHDL d'un registre tampon .....	21

<b>3-3. Registre mémoire</b>	<b>21</b>
3-3.1. Bus de données bidirectionnel	21
3-3.2. Structure d'un registre mémoire	22
3-3.3. Description VHDL	23
<b>3-4. Les registres à décalage</b>	<b>24</b>
3-4.1. Registre serial IN serial OUT	25
<i>Description VHDL</i>	25
3-4.2. Registre serial IN parallel OUT	26
<i>Description VHDL</i>	27
3-4.3. Registre parallel IN serial OUT	27
<i>Description VHDL</i>	28
3-4.4. Registre parallel IN parallel OUT	29
<i>Description VHDL</i>	29

## **Chapitre 4 Les machines d'états synchrones** **31**

<b>4-1. Structure interne d'une machine d'états</b>	<b>33</b>
<b>4-2. Le graphe des états (diagramme de transition)</b>	<b>33</b>
4-2.1. Table de vérité des sorties en fonction de l'état	35
<b>4-3. Du graphe des états à la description VHDL</b>	<b>35</b>
4-3.1. Codage des états	35
4-3.2. Machines de Moore	36
4-3.3. Machines de Mealy	37
4-3.4. Machines de Mealy à sorties resynchronisées	37

## **Chapitre 5 Les Compteurs** **39**

<b>5-1. Les Compteurs Asynchrones</b>	<b>39</b>
5-1.1. Modulo	40
5-1.2. Division de la fréquence	40
5-1.3. RETARD DE PROPAGATION DANS LES COMPTEURS ASYN- CHRONES	40
5-1.4. Compteurs asynchrones modulo $< 2N$	41
<b>5-2. COMPTEURS SYNCHRONES (PARALLÈLES)</b>	<b>42</b>
<b>5-3. COMPTEURS PRÉRÉGLABLES</b>	<b>44</b>
<b>5-4. Description d'un compteur synchrone en VHDL</b>	<b>44</b>

## Chapitre 1

# *LES BASCULES*

---

Nous avons étudié jusqu'ici des circuits logiques combinatoires dont les sorties, à un instant donné, ne dépendent que de l'état des valeurs présentes sur les entrées. Toute condition antérieure n'a aucun effet sur les valeurs actuelles des sorties, parce que les circuits combinatoires n'ont pas de mémoire. Dans la majorité des systèmes numériques, on retrouve une combinaison de circuits combinatoires et de dispositifs à mémoire.

La section combinatoire est alimentée par des signaux d'entrée externes et par les sorties des dispositifs à mémoire. Le circuit combinatoire agit sur ces entrées pour produire diverses sorties, certaines servant à déterminer les valeurs binaires stockées dans les éléments de mémoire. La sortie de certains de ces éléments de mémoire revient comme entrée des circuits logiques de la partie combinatoire. Ceci est une indication que les sorties externes d'un système numérique dépendent autant des entrées externes que des informations mémorisées dans d'autres sections. On appelle cela un système séquentiel.

## 1-1 Définition du système séquentiel

---

Un système est dit séquentiel, si à un même vecteur d'entrée, il fait correspondre plusieurs vecteurs de sortie différents. Chaque vecteur de sortie dépendra alors non seulement du vecteur d'entrée à l'instant  $t$ , mais aussi des précédents, ce qui introduit la notion de séquence d'entrée.

L'effet mémoire est typique des systèmes séquentiels, l'élément de mémorisation le plus important est la bascule D, constituée d'un ensemble de portes logiques. Même si, en soi, une porte logique ne retient pas de donnée, il est possible d'en raccorder quelques-unes ensemble afin d'obtenir le stockage d'une information. Il existe différentes façons de monter les portes pour obtenir ces bascules.

## 1-2 Bascule R-S en portes NAND

---

La bascule la plus élémentaire est construite au moyen de deux portes NAND ou de deux portes NOR. La version NON ET, appelé élément de mémoire en portes NAND ou bascule R-S (dans certains cas on l'appelle aussi bascule SC), est montrée à la figure 1- 1b). Les deux portes NAND sont rétro-couplées, de sorte que la sortie de la porte NAND 1 est connectée à l'une des entrées de la porte NAND 2, et vice versa. Les sorties, appelée  $Q$  et  $\bar{Q}$  respectivement, sont les sorties de l'élément de mémoire. Dans des conditions normales, une sortie est toujours l'inverse de l'autre. Les entrées de l'élément de mémoire sont désignées S (d'après SET) et R (d'après RESET). Les entrées S et R se trouvent normalement toutes les deux au niveau HAUT, et l'on doit momentanément en porter une au niveau BAS pour changer l'état de sortie de la bascule. Un élément de mémoire en NAND possède deux états stables possibles quand  $S = R = 1$ . L'élément de mémoire en NAND peut se mettre sous la forme d'une table de vérité (figure 1- 1a);

- a.  $S = R = 1$ ; cette condition correspond à l'état normal de repos et elle n'affecte pas l'état de sortie de la bascule. Les sorties demeurent dans l'état qu'elles occupaient avant l'application de cette condition d'entrée.
- b.  $S = 0, R = 1$ ; cette condition entraîne toujours la sortie dans l'état 1 où demeure même après le retour de S au niveau HAUT. On dit que c'est la condition de mise à 1 de la mémoire (SET).
- c.  $S = 1, R = 0$ ; cette condition entraîne toujours la sortie dans l'état 0 où demeure même après le retour de C au niveau HAUT. On dit que c'est la condition de mise à 0 de la mémoire (RESET).
- d.  $S = R = 0$ ; cette condition est équivalente à vouloir mettre la mémoire fois à 1 et à 0, ce qui donne lieu à des résultats ambigus. En fait les deux sorties sont à l'état HAUT simultanément. Cette

condition ne doit jamais servir.

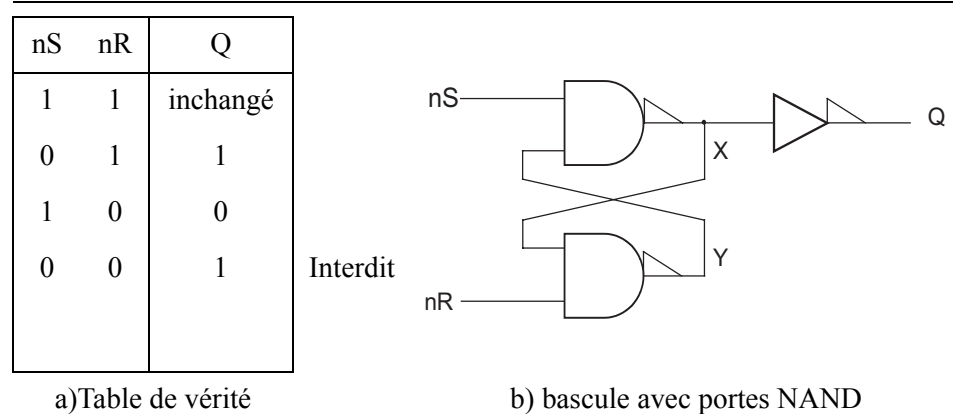


Figure 1- 1 : Table de vérité et schéma logique d'une bascule R-S NAND

### 1-3 Bascule R-S en portes NOR

Deux portes NOR rétro-couplées constituent une mémoire R-S (bascule R-S). Un tel montage, illustré à la figure 1- 2b), est analogue à celui d'une mémoire en NAND, sauf que les sorties  $Q$  et  $\bar{Q}$  sont maintenant interverties.

L'étude du fonctionnement d'une mémoire en NOR se développe de manière tout à fait identique à celle de la mémoire en NAND. Les résultats sont donnés sous forme d'une table de vérité à la figure 1- 2a) et résumés ci-après:

- $S = R = 0$ ; cette condition représente l'état normal de repos de la mémoire en NOR et ne modifie en rien l'état de sortie.  $Q$  et demeurent dans l'état qu'elles occupaient avant l'arrivée de l'impulsion d'entrée.
- $S = 1, R = 0$ ; cette condition a toujours pour effet de mettre  $Q$  à 1, état qui ne change pas même quand  $S$  revient à 0.
- $S = 0, R = 1$  ; cette condition a toujours pour effet de mettre  $Q$  à 0 , état qui ne change pas même quand  $C$  revient à 0.
- $S = R = 1$ ; cette condition est équivalente à vouloir mettre la mémoire à 1 et à 0 en même temps et produit  $Q = = 0$ . Si les deux entrées sont ramenées simultanément à 0, l'état de sortie résultante est imprévisible. Il ne faut jamais se servir de cette condition d'entrée.

L'élément de mémoire en NOR fonctionne en tous points comme la mémoire en NAND, à l'exception des entrées  $S$  et  $R$  qui, maintenant, sont vraies au niveau HAUT plutôt qu'au niveau BAS, et de l'état normal de repos qui est  $S = R = 0$ .  $Q$  sera mis à 1 par une impulsion de niveau HAUT

appliquée sur S et sera mis à 0 par une impulsion, toujours de niveau HAUT, sur R.

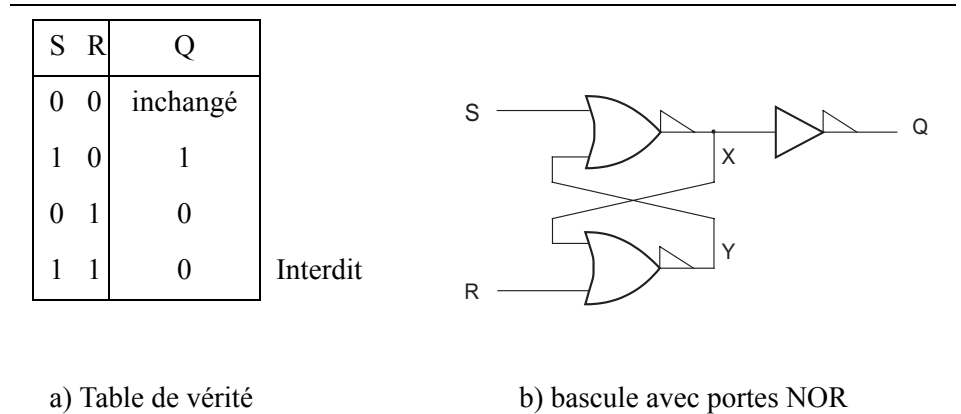


Figure 1- 2 : Table de vérité et schéma logique d'une bascule R-S NOR

### 1-4 Bascule R-S avec enable

Les bascules R-S étudiées précédemment sont sensibles aux changement sur les entrées S et R, ceci à n'importe quel moment. Cependant il est très facile de modifier un de ces circuits pour le rendre ses entrées sensibles uniquement lorsque une troisième entrée enable (C) est active. Une telle bascule R-S avec enable est montrée dans la figure 1- 3. Comme on peut le voir dans la table de vérité, ce circuit se comporte comme une simple bascule R-S lorsque l'entrée C (enable) est à 1, et mémorise son état lorsque C est à 0. En effet si l'entrée C est à 0, on retrouve 1 à la sortie des deux porte NAND, ce qui équivaut à l'état mémorisation sur une bascule R-S en porte NAND. Dans le cas où C vaut 1, la porte NAND agit comme un simple inverseur, il nous reste donc une bascule R-S NAND avec entrées inversées.

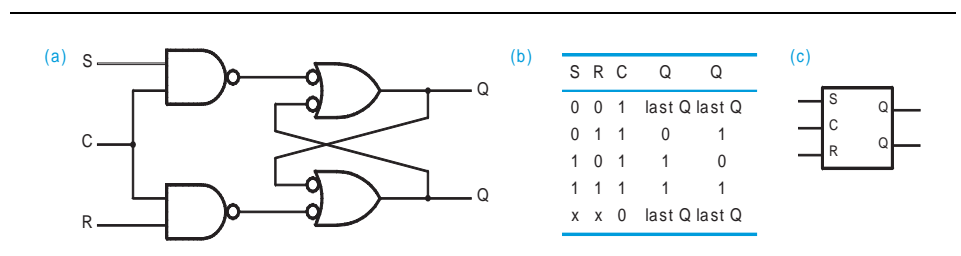


Figure 1- 3 : Bascule R-S avec enable: (a) circuit logique; (b) table de vérité; (c) symbole logique

La figure 1- 4 montre le fonctionnement typique d'une bascule R-S avec enable pour des signaux données. Si les deux entrées S et R sont simultanément à l'état logique haut lorsque C passe de 1 à 0, l'état suivant est imprévisible, et la sortie peut se trouver dans un état métastable.



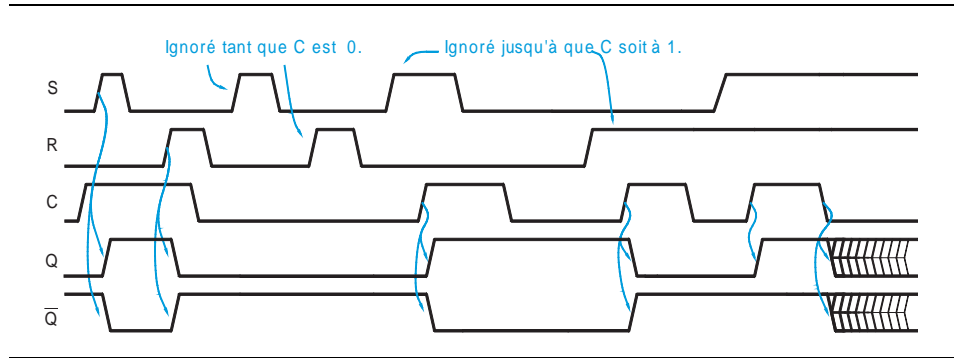


Figure 1- 4 : Fonctionnement typique d'une bascule R-S avec enable

### 1-5 L'élément mémoire D (D Latch)

L'élément mémoire D plus souvent nommé D Latch est une bascule R-S avec enable sur laquelle on a simplement ajouté un inverseur entre les entrées S et R, il ne reste donc plus que l'entrée appelée D. Ceci nous supprime la possibilité de mettre S et R simultanément à 1 en même temps, ce qui élimine les problèmes de métastabilité de la bascule R-S avec enable. On peut voir le schéma logique d'un D latch à la figure 1- 5(a).

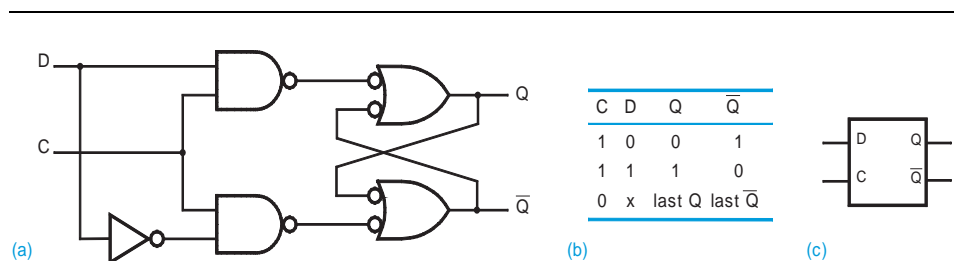


Figure 1- 5 : D latch: (a) schéma logique ; b) table de vérité ; c) symbole logique.

On peut voir dans figure 1- 6 ci-dessous le fonctionnement d'un D Latch pour des variations d'entrées données. Le fonctionnement est très simple, tant que C est à l'état logique haut, ce qui est sur l'entrée D se retrouve sur la sortie Q, et lorsque C est à l'état logique bas, la sortie mémorise le dernier état, même en cas de changement sur l'entrée D.

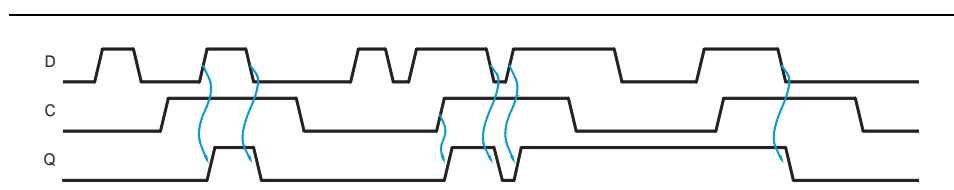
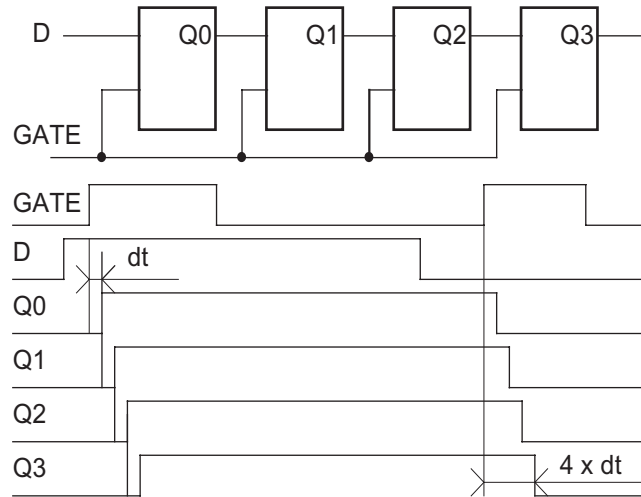


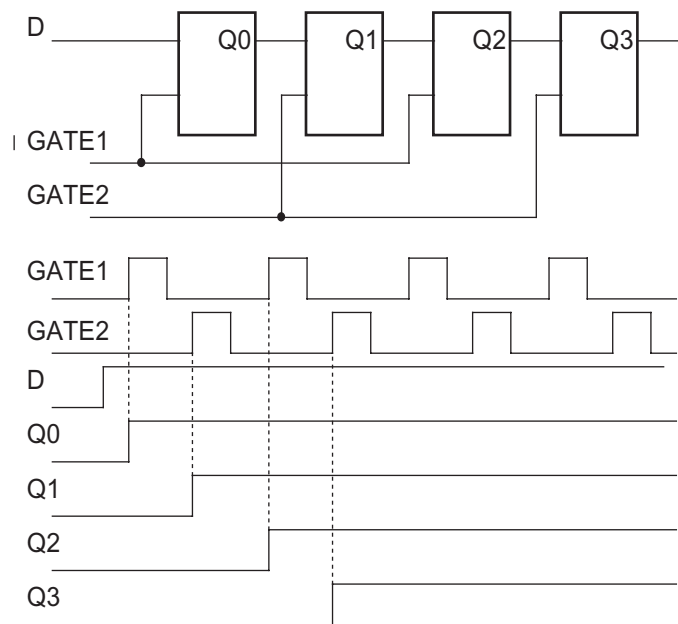
Figure 1- 6 : Fonctionnement d'un D Latch pour des variations d'entrées données

## 1-6 La bascule maître-esclave.

Pour réaliser un registre à décalage (chaîne de bascules dans laquelle l'information se propage d'un élément à l'autre à chaque commande), une bascule type latch n'est pas suffisante. L'ouverture des latches entraînera un décalage de l'information, qui ne serait contrôlé que par le temps d'ouverture et le temps de propagation dans les latches, et non d'une unité à chaque action sur la commande.

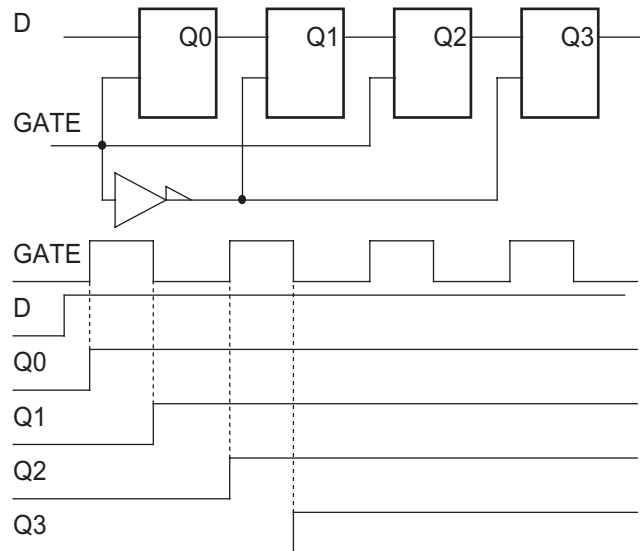


Pour palier à cet inconvénient, nous sommes amenés à utiliser des latches avec des entrées de commandes qui ne reçoivent pas le même signal de commande.

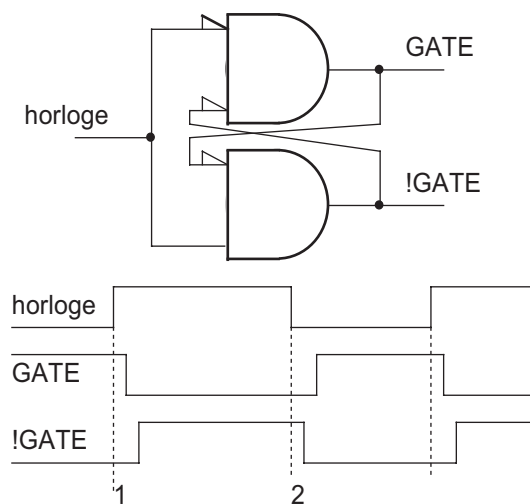


Il est encore nécessaire que la commande GATE soit maintenue au niveau haut pendant un temps supérieur à  $dt$ , si  $dt$  est le temps de réaction du

latch. Cette solution présente l'inconvénient de nécessiter deux signaux d'horloge. Une amélioration peut y être apportée en utilisant l'horloge et son inverse. Dans ce cas, nous sommes conduits à la structure dite maître/esclave en combinant les bascules deux à deux. Il est toutefois nécessaire que le temps de propagation de la porte non soit inférieur au temps de réaction de la bascule, ce qui est le cas, car la commutation de la bascule nécessite le temps de réaction de trois portes chaînées.



Un moyen d'éviter le chevauchement de GATE et !GATE consiste à utiliser le schéma ci-après. Dans ce montage, au temps 1 la montée de l'horloge provoque la descente de GATE, puis la montées de !GATE. De même au temps 2 !GATE descend avant la montée de GATE. Les deux signaux GATE et !GATE ne sont jamais ensemble au niveau haut.

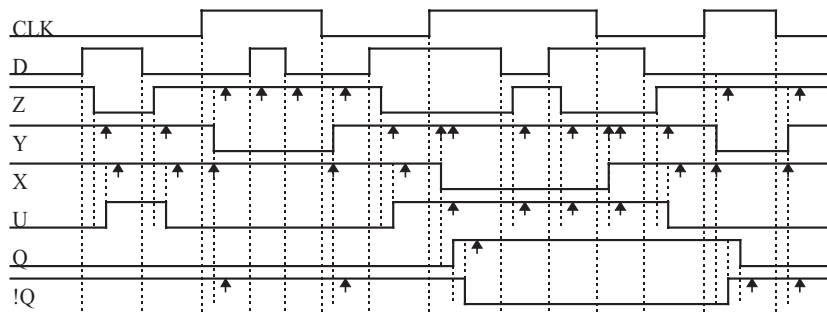
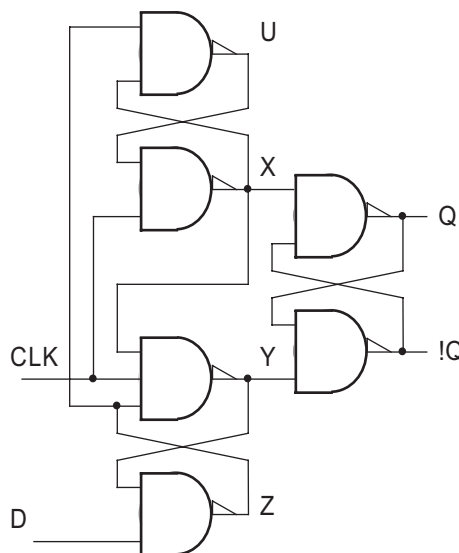


Cette bascule s'appelle maître/esclave ou flip-flop du nom des bruits des relais qui la matérialisait autrefois.

Dans la bascule maître-esclave, les entrées sont actives pendant que l'horloge est au niveau haut, et les sorties ne sont modifiées qu'au flanc descendant de l'horloge (nous dirons aussi au flanc négatif). La valeur des sorties juste après le flanc descendant de l'horloge dépend de la valeur des entrées présentes avant cette transition.

### 1-7 La bascule bistable déclenchable sur un flanc "edge-triggered"

Les bascules maîtres-esclaves sont sensibles aux impulsions qui peuvent apparaître sur leurs entrées pendant le temps d'ouverture du maître. Ce comportement non souhaitable est évitable en utilisant des bascules edge-triggered dont le changement d'état intervient sur le flanc de l'horloge et non plus sur le niveau.



Exemple 1- 1 : Bascule D type 74xx74.

Dans de telles bascules, l'échantillonnage de l'entrée n'a pas lieu sur un niveau de l'horloge, mais lors du changement d'état de cette dernière, ce qui

rend la bascule beaucoup moins sensible aux parasites. La conception et la réalisation de ces bascules est complexe, et nous ne la traiterons pas ici.

## 1-8 Signal d'horloge et bascules synchrones

Les circuits numériques peuvent fonctionner de façon synchrone ou asynchrone. Dans les systèmes asynchrones, la sortie des circuits logiques peut changer d'état à tout moment quand une ou plusieurs entrées changent. Un système asynchrone est difficile à concevoir et à déboguer.

Par contre dans un système synchrone, le moment exact où la sortie change d'état est commandé par un signal que l'on appelle couramment signal d'horloge. Ce signal est généralement un train d'ondes rectangulaires ou carrées, comme ceux de la figure 1- 7. Le signal d'horloge est habituellement distribué à tous les étages du système, de sorte que la plupart des sorties du système, sinon toutes, changent d'état seulement quand le signal d'horloge effectue une transition. Ces transitions, appelées fronts ou flancs. Quand le signal d'horloge passe de 0 à 1, on parle du front montant (transition positive), quand il passe de 1 à 0, on parle de front descendant (transition négative).

La majorité des systèmes numériques existants sont des machines synchrones du fait que les circuits synchrones sont plus simples à concevoir et à dépanner. Leurs sorties ne peuvent changer qu'à des instants précis bien connus. Autrement dit, tous les changements sont synchronisés avec les transitions du signal d'horloge.

La synchronisation orchestrée par des signaux d'horloge est réalisée au moyen de bascules synchrones qui ont été réalisées pour changer d'état au moment de la transition associée à un front ou à l'autre du signal d'horloge.

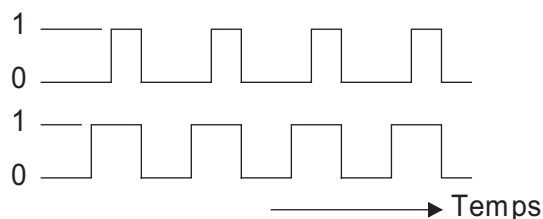
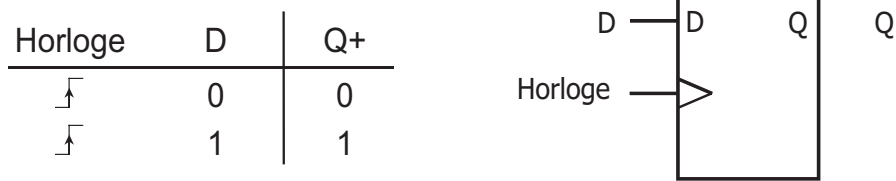


Figure 1- 7 : Signaux d'horloge

### 1-9 FLIP-FLOP D



### 1-10 BASCULE T (Toggle flip flop)

Pour réaliser une bascule T, il suffit de connecter la sortie  $\bar{Q}$  sur l'entrée D d'une bascule D, comme sur la figure 1- 8 a), ou alors si on n'a pas de sortie  $\bar{Q}$ , de relier la sortie Q sur l'entrée d'un inverseur, dont la sortie sera connectée sur l'entrée D. Le fonctionnement de la bascule T est très simple: A chaque flanc montant du signal d'horloge, la sortie Q change d'état. Les formes d'ondes de la figure 1- 8 c) illustrent ce fonctionnement.

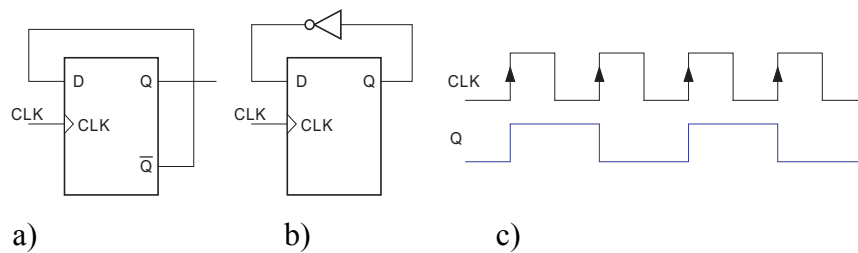


Figure 1- 8 : Réalisations d'une bascule T (toggle flip-flop) ; c) Formes d'ondes sur une bascule T.

### 1-11 Entrées asynchrones

Dans les bascules synchrones précédentes, nous avons parlé d'entrées de commande pour les entrées S, C, et D. Ces entrées sont également qualifiées de synchrones étant donné que la sortie de la bascule est synchronisée par le signal d'horloge. Comme nous l'avons vu, les entrées de commande synchrones sont utilisées concurremment à un signal d'horloge qui déclenche la réponse de la bascule.

La majeure partie des bascules synchrones possèdent en plus des entrées asynchrones qui agissent indépendamment des entrées synchrones et du signal d'horloge. On a recours à de telles entrées pour forcer en tout temps la remise à 1 ou à 0 de la bascule, quelles que soient les conditions des entrées. Une autre façon de présenter ces entrées est de dire que ce sont

des entrées prioritaires qui imposent un état à la bascule malgré les commandes lancées par les autres entrées.

La figure 1- 9 illustre une bascule D munie d'une entrée asynchrone désignée CLR. C'est une entrée vraie au niveau HAUT. La table à droite de ce symbole résume la réaction de la sortie de la bascule.

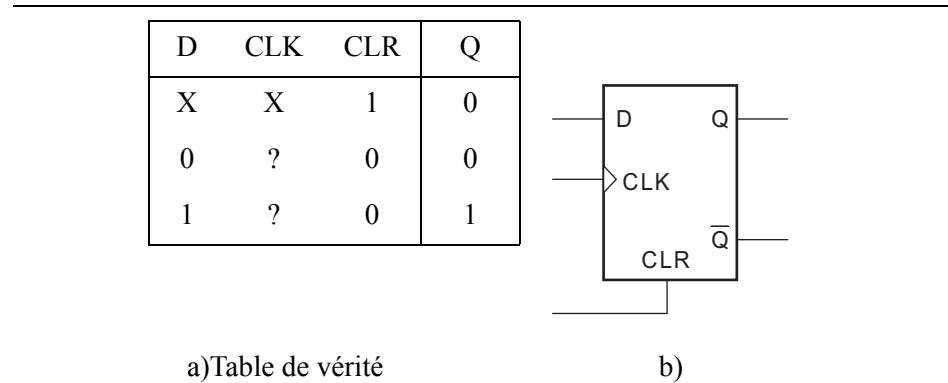


Figure 1- 9 : Table de vérité et symbole logique d'une bascule D avec CLEAR asynchrone actif haut.

Il est important de se rendre compte que les entrées asynchrones sont des niveaux de tension continue (CC). Cela veut dire que si CLR est gardé au niveau 1, la bascule restera dans l'état  $Q = 0$  quoiqu'il arrive aux autres entrées. Le plus souvent, toutefois, on applique momentanément à ces entrées une impulsion pour remettre la bascule à 1 ou à 0.

De nombreuses bascules synchrones fabriquées sous forme de circuits intégrés possèdent deux entrées asynchrones, d'autres n'en ont qu'une. Certaines bascules ont des entrées asynchrones qui sont vraies au niveau HAUT plutôt qu'au niveau BAS.

### 1-11.1 Désignation des entrées asynchrones

Dans leurs fiches techniques, les fabricants de CI emploient différentes désignations pour les entrées asynchrones, et malheureusement aucune n'a été normalisée. Voici les désignations que vous êtes susceptibles de rencontrer dans les schémas et les fiches techniques anglaises des fabricants. Pour la mise à l'état 1 : SET, PRESET (PRE), pour la mise à l'état 0 : CLEAR (CLR), RESET.

## 1-12 Considérations sur la synchronisation des bascules

Les fabricants de bascules intégrées spécifient quelques paramètres et caractéristiques de synchronisation importants qu'il importe de prendre en considération avant d'utiliser une bascule dans un circuit pratique.

### 1-12.1 Temps de stabilisation (setup time) et temps de maintien (hold time)

Deux exigences de synchronisation doivent être respectées pour qu'une bascule synchrone réponde correctement à ses entrées de commande lorsque arrive le front déclencheur de CLK. Ces exigences sont représentées sous forme graphique sur la figure 1- 10 pour une bascule déclenchée par un front montant.

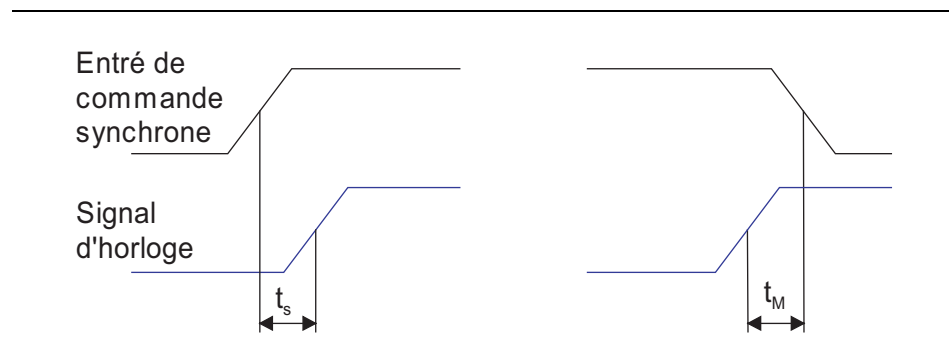


Figure 1- 10 : Les entrées de commande doivent rester stables pendant a) un temps  $t_S$  avant la transition active de l'horloge, et b) un temps  $t_M$  après la transition active de l'horloge.

- Le temps de stabilisation (ou  $t_{setup}$  dans les databook),  $t_S$ , est l'intervalle qui précède immédiatement le front déclencheur du signal d'horloge, pendant lequel l'entrée synchrone doit être gardée au niveau approprié. Les fabricants de CI spécifient généralement la durée de stabilisation minimale admissible. Si on ne respecte pas ce temps, il n'est pas garanti que la bascule répondra correctement à l'arrivée du front.
- Le temps de maintien (ou  $t_{hold}$  dans les databook),  $t_M$ , est l'intervalle qui suit immédiatement le front déclencheur du signal d'horloge pendant lequel l'entrée synchrone doit être gardée au niveau approprié. Les fabricants de CI spécifient généralement le temps minimal acceptable. Si on ne respecte pas ce temps, la bascule ne sera pas déclenchée correctement.

D'après ce qui vient d'être dit, on voit que l'entrée de commande doit rester stable (inchangée) pendant une durée égale à la somme du temps  $t_S$ , qui précède le front déclencheur, et du temps  $t_M$ , qui suit ce même front. Les bascules de CI ont des temps  $t_S$  et  $t_M$  de l'ordre des nanosecondes.

### 1-12.2 Temps de propagation

Chaque fois qu'un signal doit changer l'état d'une bascule, on observe un retard entre le moment où le signal est appliqué et le moment où le changement apparaît à la sortie. Une illustration des retards de propagation affectant la réponse à un front montant de signal d'horloge nous est fournie à la figure 1- 11. Notez que ces retards sont mesurés entre les points à mi-



hauteur (50 %) des formes d'ondes d'entrée et de sortie. Les mêmes genres de retards se produisent en réponse à des signaux placés sur les entrées asynchrones (SET et CLR). Sur les fiches techniques des fabricants, on trouve généralement les retards de propagation affectant la réponse à toutes les entrées, ainsi que les valeurs maximales de  $t_{PLH}$  et  $t_{PHL}$ .

Les bascules modernes dans les CI ont des retards de propagation sont de l'ordre de quelques nanosecondes. Les valeurs de  $t_{PLH}$  et  $t_{PHL}$  ne sont pas égales et augmentent proportionnellement avec le nombre d'étages logiques pilotés par la sortie Q. Les retards de propagation dans les bascules jouent un rôle important dans les circuits logiques.

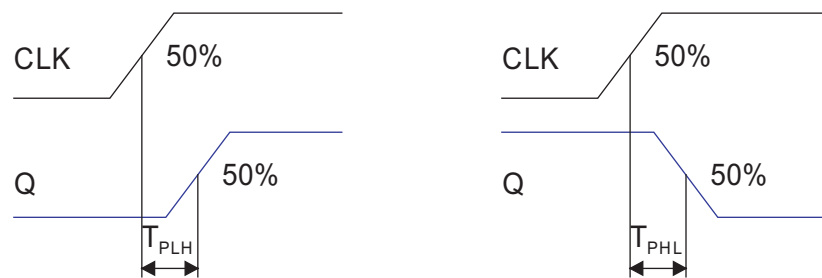


Figure 1- 11 : Retards de propagation dans une bascule.

### 1-13 Description d'une bascule D avec reset asynchrone en VHDL

Dans tous les systèmes séquentiels, on doit utiliser un process, et pour pouvoir identifier un flanc du signal d'horloge, on utilise la fonction `Rising_Edge`, celle-ci renvoie une valeur booléenne TRUE lors d'un flanc montant (`Falling_Edge` pour flanc descendant) du signal d'horloge.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity DFF is
    Port ( Clk      : in  std_logic;
          nRreset  : in  std_logic;
          D        : in  std_logic;
          Q        : out std_logic);
end DFF;

architecture Comport of DFF is
    signal Reset : Std_Logic;
begin
    --adaptation polarite
    Reset <= not nReset;

    process(Clk, Reset)

```

```
begin
  if Reset = '1' then           --reset asynchrone
    Q <= '0';
  elsif Rising_edge(Clk) then  -flanc montant du signal
    Q <= D;
  end if;
end process;

end Comport;
```

---

Exemple 1- 2 : Description d'une bascule D avec reset asynchrone en VHDL

### Traduction:

- Si nReset est actif alors la sortie Q prend l'état 0 (si la condition est vraie, on sort du process, donc le reset est prioritaire et asynchrone).
- Sinon (sous-entendu reset inactif) si on a un flanc montant sur Clk, alors la sortie Q prend la valeur de l'entrée d.
- Si aucune condition est vraie, rien ne change.

Ceci est bien le fonctionnement de la bascule D.

**Remarque importante:** Comme déjà mentionné précédemment, en logique combinatoire, on doit spécifier tous les cas possibles sur les entrées, sous peine de voir le synthétiseur introduire des éléments mémoire non désirés. En logique séquentielle par contre, on souhaite obtenir un élément mémoire. Donc on ne spécifie pas les cas où le signal d'horloge n'est pas un flanc montant. Dans ce cas le signal Q n'est pas modifié et nous obtenons bien une bascule D.

## 1-14 Synchronisation des entrées externes asynchrones

---

Les systèmes numériques travaillent toujours de façon synchrone, c'est pourquoi il est nécessaire de synchroniser les entrées externes asynchrones avec le signal d'horloge (Clk). En effet un changement de valeur du signal externe peu se produire à n'importe quel instant, c'est pour quoi on utilise des bascules D mises en série pour synchroniser une entrée externe. La figure 1- 12 nous monte une synchronisation à trois bascules D.

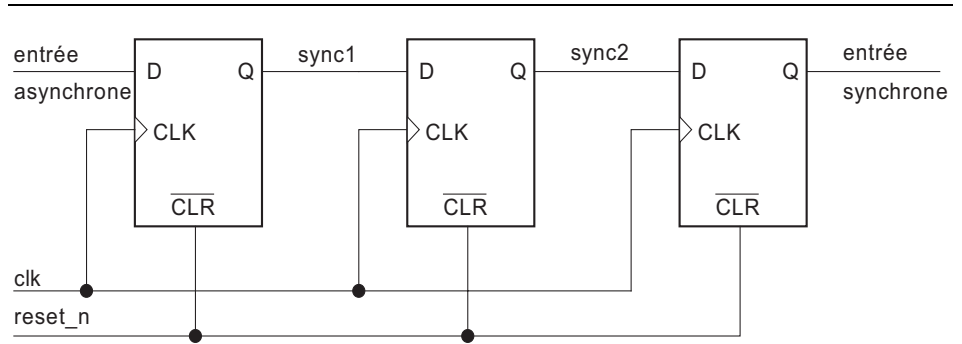


Figure 1- 12 : Synchronisation d'une entrée avec 3 bascules D

On utilise deux voir trois bascule pour supprimé le phénomène de métastabilité sur la première bascule dans le cas où le changement sur l'entrée asynchrone surviendrait exactement au moment du flanc montant du clock. Un exemple est donnée dans le chronogramme ci-dessous :

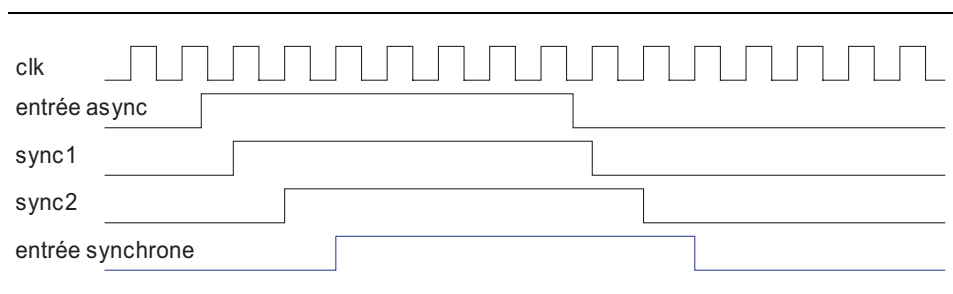


Figure 1- 13 : Chronogramme de la synchronisation d'un signal





ment lorsque tgl\_sync2 est différent de tgl\_sync3, c'est à dire lorsque la transition sur tgl\_async arrive sur tgl\_sync2. Ci dessous le chronogramme montrant en détail chaque signal:

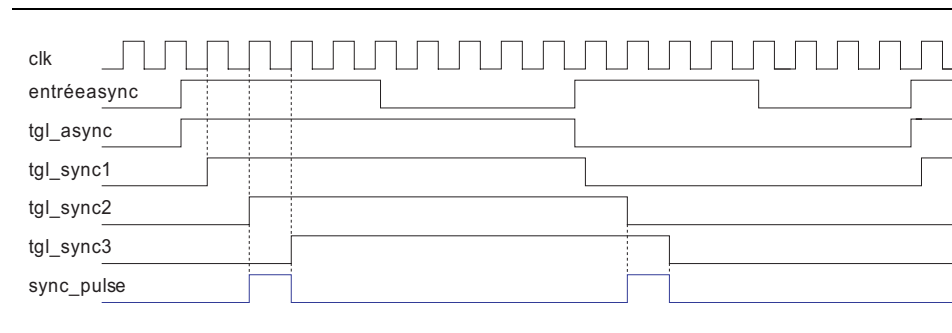


Figure 2- 2 : Chronogramme de la synchronisation et détection de flanc d'un signal externe