Cours : *Intro à la logique séquentielle : mémorisation*
Public : *IUT Informatique*
Auteur : *P. Pinault*

Introduction à la logique séquentielle : la mémorisation

La logique combinatoire que nous avons précédemment étudiée est telle que dès lors qu'une entrée est modifiée, la sortie est affectée par cette modification.

Dans de nombreuses applications, il est nécessaire de maintenir un signal lu précédemment, utile à un traitement ultérieur et susceptible d'évoluer entre temps.

Par exemple, l'utilisation d'un télé rupteur de couloir commandant un éclairage n'entraîne pas son extinction alors que celui-ci est relâché. Le système mémorise donc une commande d'éclairage jusqu'à recevoir un nouvel ordre.

La mémoire des systèmes logiques est basée sur des bascules ou bistables ou encore registres (mémoires élémentaires) associées entre elles.

On distingue deux familles de bascules :

- Celles dont la mémorisation est commandée par des niveaux, appelées en général bistables ou latch en anglais. Les bistables RS et D transparents en sont les principaux représentants.
- Celles dont l'état est défini par des fronts d'horloge, appelées souvent bascules synchrones ou simplement bascules ou Flip-Flop en anglais. Ces bascules sont le fondement de la logique séquentielle vue au chapitre suivant.

1. Les bistables RS

Les bistables RS sont à la base de tous les éléments de mémorisation que nous verrons par la suite. Il s'agit d'un montage utilisant deux portes NAND et capable de mémoriser un niveau logique choisi. Son schéma est le suivant :

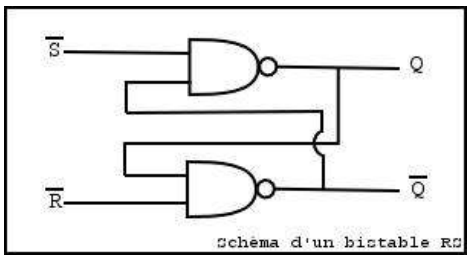
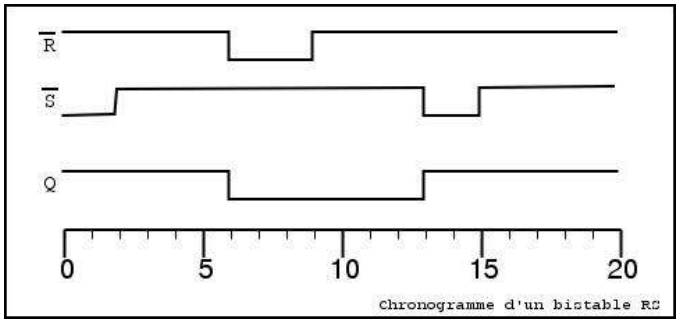


Schéma d'un bistable RS



Chronogramme d'un bistable RS

En entrée, les signaux Set et Reset, actifs à l'état bas, commandent le système.

- Lorsque Set devient actif, le système change d'état et Q prend pour valeur 1.
- Lorsque Set devient inactif, le système conserve son état et Q reste à 1.
- Reset à l'effet inverse sur le système. Les deux signaux de contrôle ne peuvent être actifs en même temps sans quoi les sorties ne seraient plus complémentaires.

La table de vérité de ce circuit est la suivante : (on note 't' l'instant avant modification des entrées).

/S	/R	Q_{t+1}
0	0	NA
0	1	1
1	0	0
1	1	Q_t

Cours : *Intro à la logique séquentielle : mémorisation*
Public : *IUT Informatique*
Auteur : *P. Pinault*

2. Les bascules D transparentes

Les bistables RS sont tout à fait utilisables pour la mémorisation, toutefois, ils nécessitent la connaissance de l'état à mémoriser de sorte à passer la bonne commande Set ou Reset. Plus généralement, le besoin est la mémorisation d'une valeur inconnue à partir d'un instant donné.

Il a donc été créé à partir d'un bistable RS, un système ne nécessitant qu'une seule entrée pour fixer le niveau à mémoriser. Ce système a en outre un second signal permettant la commande de mémorisation.

Dans un premier temps, voyons comment a été modifié le bistable RS pour le rendre synchrone :

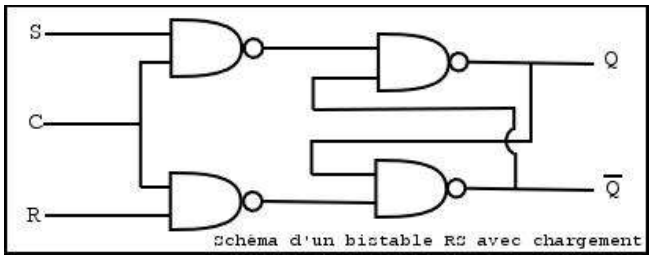
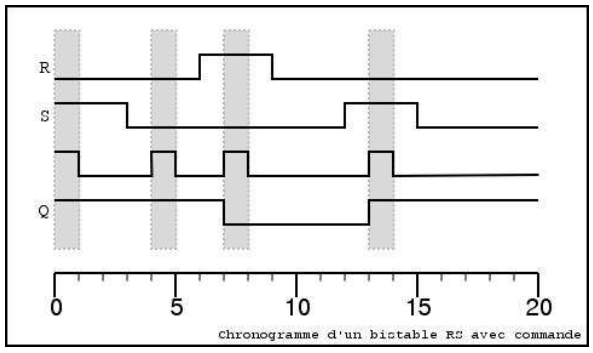


Schéma d'un bistable RS avec chargement



Chronogramme d'un bistable RS avec commande

Les ordres Set et Reset ne sont alors pris en compte qu'à la réception d'une commande *C*. Notez que les signaux Set et Reset sont rendus actifs à l'état haut.

Enfin, il est possible d'ajouter un inverseur entre Set et Reset pour n'avoir plus qu'un seul signal d'entrée. Le bistable ainsi créé est appelé bascule D transparente. Nous noterons que dans un tel montage, il n'y a plus de combinaisons d'entrées invalides.

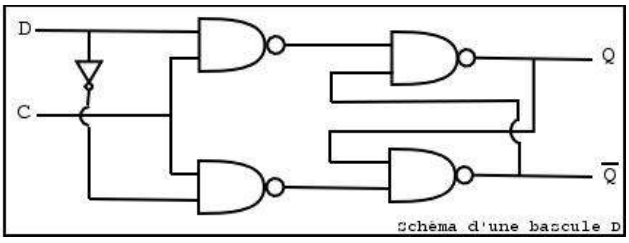
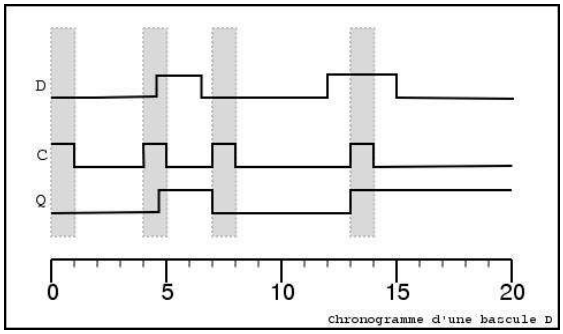


Schéma d'une bascule D



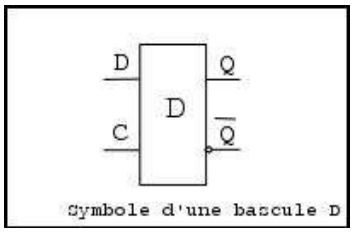
Chronogramme d'une bascule D

Le bistable va donc mémoriser la valeur lue alors que son entrée de commande C est active. La table de vérité de ce composant est la suivante :

D	**C**	$\mathbf{Q_{t+1}}$
0	0	Q_t
1	0	Q_t
0	1	0
1	1	1

Cours : *Intro à la logique séquentielle : mémorisation*
Public : *IUT Informatique*
Auteur : *P. Pinault*

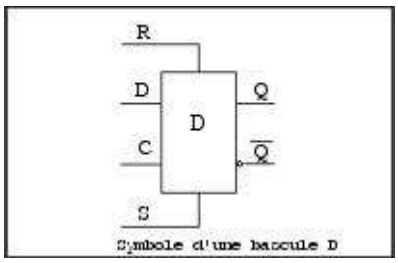
Cette bascule D est couramment représentée par le symbole suivant :



Symbole d'une bascule D

Cette bascule peut avoir d'autres signaux secondaires asynchrones : Set et Reset. Ils permettent respectivement la mise à 1 ou à 0 de la bascule. Ils permettent un forçage de la bascule de façon asynchrone, c'est à dire sans délai, et éventuellement une initialisation au démarrage du système. Lorsque sur une bascule, le signal C n'est pas utilisé, on obtient une bascule fonctionnant comme un bistable RS.

Le symbole et la table de vérité de cette bascule plus complète sont les suivants :



Symbole d'une bascule D

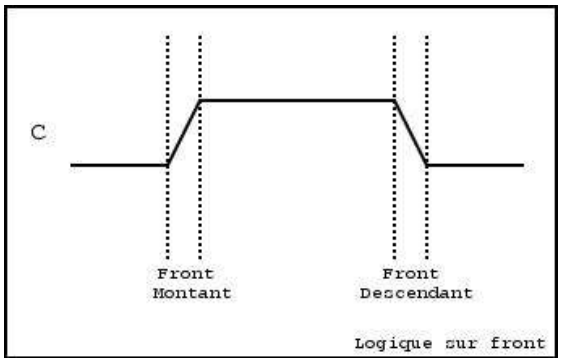
D	**C**	**R**	**S**	Q_{t+1}
0	0	0	0	Q_t
1	0	0	0	Q_t
0	1	0	0	0
1	1	0	0	1
X	X	1	0	0
X	X	0	1	1

Cours : *Intro à la logique séquentielle : mémorisation*
Public : *IUT Informatique*
Auteur : *P. Pinault*

IV.3. Logique sur front

Dans le cas d'un bistable D, la sortie Q copie le niveau de D durant toute la période d'activité du signal C. Dès que C devient inactif, et aussi longtemps qu'il le reste, le dernier niveau est mémorisé. Ce comportement est celui de la mémoire statique élémentaire, répandue dans tous les systèmes logiques.

Un comportement plus évolué est parfois souhaitable. Le fait que Q puisse changer lorsque C est actif peut être gênant dans certains cas. Pour modifier ce comportement, la bascule synchrone a été définie. Dans une telle bascule la sortie Q correspond à l'état de D au moment précis où C est passé de l'état inactif à l'état actif. Q ne change pas tant que C ne change pas, que le niveau soit 0 ou 1. On parle donc de logique sur front car seul un front d'horloge, et non pas un niveau, a une influence sur l'état du système.



Logique sur front

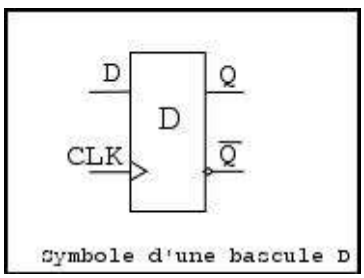
Pour des raisons de testabilité des circuits, le signal C est généralement issu d'une horloge. Ce signal sera donc plus généralement appelé H ou encore CLK.

La table de vérité de la bascule D est légèrement différente :

D	CLK	Q_{t+1}
0	↓	0
1	↓	1
X	0	Q_t
X	1	Q_t

Q - représentez le chronogramme de Q engendré par :
(D,CLK):(1,0),(0,0),(0,1),(1,1)(1,0),(1,1),(1,1),(0,0),(0,1),(0,0),(1,0),(1,1)

Alors, le symbole utilisé est légèrement différent : un triangle au niveau de l'entrée CLK vient en indiquer un fonctionnement sur front.



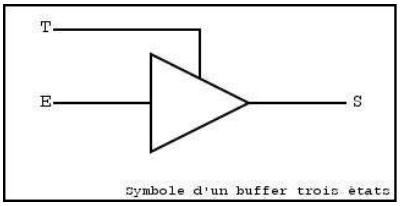
Symbole d'une bascule D

IV.5. L'état haute impédance

Dans le système vu au paragraphe précédent, nous utilisons des pattes du composant comme entrées et d'autres comme sorties. Dans beaucoup de circuits, une seule patte est utilisée à la fois comme entrée et sortie. Or, il n'est pas possible de relier simplement les deux pattes ensembles, en effet, l'entrée ne pourrait lire une autre valeur que celle fixée par la sortie, et pire encore, un court circuit pourrait être créé.

Un système permet donc de n'activer la sortie sur la patte que lorsque cela est nécessaire. Lorsqu'elle n'est pas active, la sortie est tout simplement débranchée. Cet interrupteur commandé électriquement est appelé un buffer trois états. On parle en effet de troisième état ou de haute impédance pour identifier un état différent des niveaux 0 et 1 logique. Cet état correspond à l'état débranché. L'état haute impédance se note Z.

Le composant utilisé pour cela est appelé buffer trois états, son symbole et sa table de vérité sont les suivants :



Symbole d'un buffer trois états

T	E	S
0	0	Z
0	1	Z
1	0	0
1	1	1

Ce composant est principalement employé sur les bus de communication (ISA/PCI... par exemple). En effet, un bus est un ensemble de fils permettant l'interconnexion de plusieurs périphériques. Alors plusieurs éléments sont susceptibles de présenter des données. Pour que le système fonctionne, il est impératif qu'un seul périphérique n'écrive sur le bus à un instant donné. Les autres doivent donc être placés en haute impédance sur le bus de sorte à ne pas perturber les signaux.

Les buffers trois états sont employés à cela. L'adresse accédée sur le bus permet au périphérique de savoir qu'il est interrogé et commande sa connexion au bus (l'activation de ses buffers).

Attention : bien que ce composant soit appelé un buffer, il n'a aucun effet de mémorisation. Il faut bien distinguer ce terme d'électronique de celui employé en informatique.