

Logique séquentielle

IV.1 Logique séquentielle asynchrone et synchrone

Dans le chapitre précédent portant sur la logique combinatoire nous avons fait abstraction du temps : les signaux de sortie ne dépendaient que des états des variables d'entrée. Pour les circuits de logique séquentielle nous devons tenir compte de l'état du système. Ainsi les sorties dépendent des entrées mais également de l'état du système. Celui-ci dépend aussi des entrées. Si nous notons Q l'état d'un système séquentiel, X ses entrées et Y ses sorties, nous avons de manière générale :

$$\begin{cases} Q = f(X, Q) \\ Y = g(X, Q) \end{cases}$$

La logique séquentielle permet de réaliser des circuits dont le comportement est variable avec le temps. L'état d'un système constitue une mémoire du passé.

Lorsque les changements d'état des divers composants d'un circuit séquentiel se produisent à des instants qui dépendent des temps de réponse des autres composants et des temps de propagation des signaux on parle de logique séquentielle asynchrone. Cependant les retards peuvent ne pas être identiques pour toutes les variables binaires et conduire à certains aléas. Ceux-ci peuvent être évités en synchronisant la séquence des diverses opérations logiques sur les signaux périodiques provenant d'une horloge. La logique séquentielle est alors dite synchrone : tous les changements d'état sont synchronisés sur un signal de contrôle.

Nous commençons notre étude par celle des bascules, éléments de base des circuits séquentiels. Puis nous étudierons les registres et les compteurs.

IV.2 Les bascules

Une bascule (flip-flop) a pour rôle de mémoriser une information élémentaire. C'est une mémoire à 1 bit. Une bascule possède deux sorties complémentaires Q et \bar{Q} . La mémorisation fait appel à un verrou (latch) ou système de blocage, dont le principe de rétro-action peut être représenté de la façon suivante :

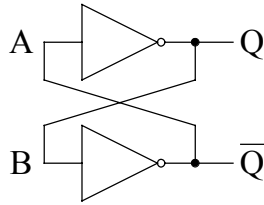


Figure 1

Nous pouvons vérifier :

$$\begin{cases} (Q = 1) \Rightarrow (B = 1) \Rightarrow (\bar{Q} = 0) \Rightarrow (A = 0) \Rightarrow (Q = 1) \\ (Q = 0) \Rightarrow (B = 0) \Rightarrow (\bar{Q} = 1) \Rightarrow (A = 1) \Rightarrow (Q = 0) \end{cases}$$

Une bascule ne peut donc être que dans deux états : "1" ($Q = 1, \bar{Q} = 0$) et "0" ($Q = 0, \bar{Q} = 1$). Les interconnexions du verrou interdisent les deux autres combinaisons : $Q = \bar{Q} = 1$ ou $Q = \bar{Q} = 0$. Ce type de circuit, qui n'a que deux états stables possibles, est encore appelé circuit bistable.

Un verrou permet de conserver un état, il nous faut maintenant savoir comment charger cet état.

IV.2.a Les bascules R-S

Les verrous les plus fréquemment rencontrés sont réalisés avec deux portes NOR ou NAND. Considérons dans un premier temps le circuit suivant :

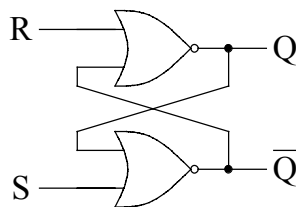


Figure 2

La table 1 donne la table de vérité correspondante. Si on applique $S = 1$ et $R = 0$ ou $S = 0$ et $R = 1$ on impose l'état de la sortie Q respectivement à 1 ou à 0, la sortie \bar{Q} prenant la valeur complémentaire. Cet état se maintient lorsque les deux entrées retournent à 0. La configuration $S = R = 1$ est à proscrire car ici elle conduit à $Q = \bar{Q} = 0$, ce qui est inconsistant logiquement avec notre définition. Mais surtout, lorsque R et S reviennent à 0, l'état $Q = \bar{Q}$ étant incompatible avec les interconnexions, l'une de ces deux sorties va reprendre l'état 1, mais il est impossible de

prédire laquelle : la configuration $S = R = 1$ conduit à une indétermination de l'état des sorties et est donc inutilisable. La représentation d'une bascule RS est donnée sur la figure 4.

S	R	Q	\bar{Q}	
0	0	Q	\bar{Q}	Sorties inchangées
1	0	1	0	Set : <u>R</u> emise à <u>U</u> n : RAU
0	1	0	1	Reset : <u>R</u> emise à <u>Z</u> éro : RAZ
1	1	0	0	A proscrire

Table 1

Considérons maintenant la bascule réalisée avec des portes NON-ET.

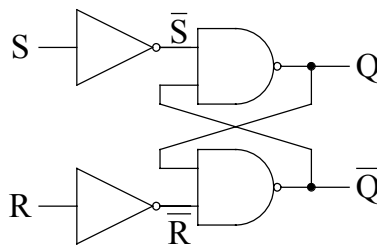


Figure 3

Il lui correspond la table de vérité suivante :

S	R	\bar{S}	\bar{R}	Q	\bar{Q}
0	0	1	1	Q	\bar{Q}
1	0	0	1	1	0
0	1	1	0	0	1
1	1	0	0	1	1

Table 2

L'utilisation des deux inverseurs sur les lignes d'entrée nous permet de retrouver une table de vérité comparable à celle de la bascule RS précédente.

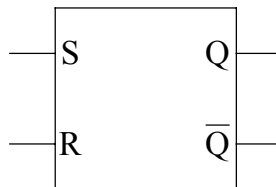


Figure 4

IV.2.b Dispositif anti-rebond

On est souvent amené à commander un état physique à l'aide d'un bouton poussoir ou d'un inverseur (par exemple les touches d'un clavier). Cependant lorsqu'on ferme un interrupteur le contact n'est pas franc instantanément et on peut observer une série de rebonds du signal avant d'obtenir la position fermée (fig. 5). Pour éviter ce défaut on réalise un montage équivalent à celui présenté sur la figure 6 faisant appel à un inverseur et une bascule.

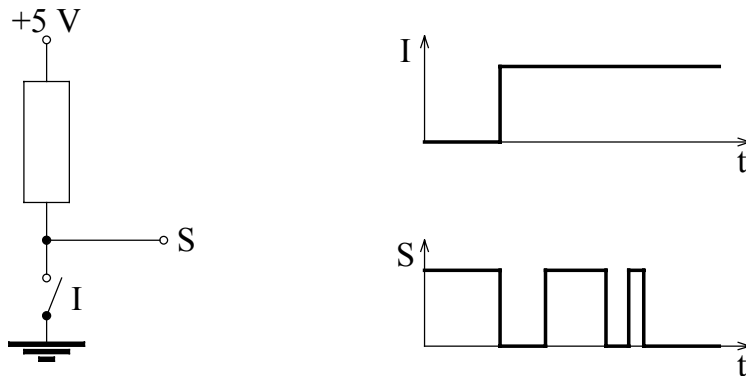


Figure 5

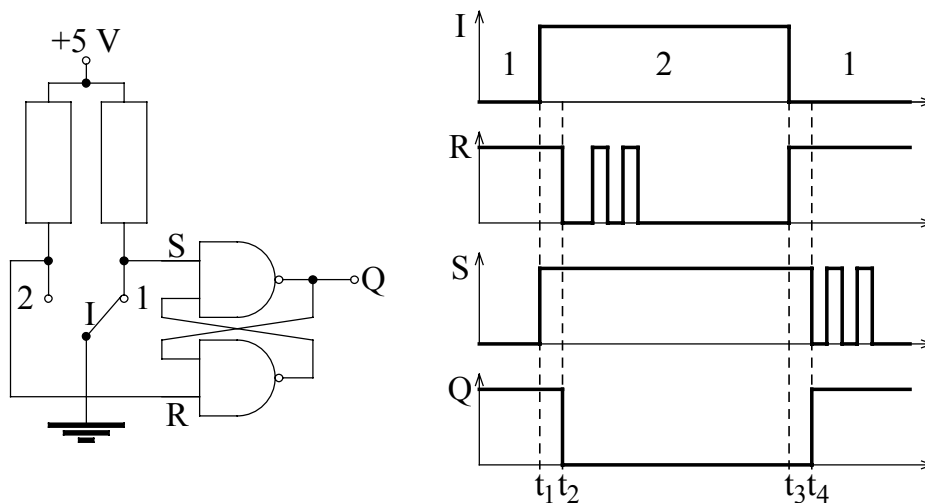


Figure 6

Supposons l'interrupteur initialement dans la position 1 comme indiqué par la figure 6 : l'entrée S est à la masse donc, en logique positive, dans l'état logique "0" et l'entrée R sous tension c'est-à-dire dans l'état logique "1". La sortie Q se trouve donc dans l'état "1". A l'instant t_1 nous basculons l'interrupteur de la position 1 à la position 2. Dès que le contact est ouvert en 1

l'entrée S se trouve sous tension et passe dans l'état logique "1". Par contre le contact en 2 peut s'établir avec un léger retard dû au mouvement du contacteur et ne pas être immédiatement franc. L'entrée R passe dans l'état "0" à t_2 , quelques rebonds peuvent suivre. A t_2 lorsque les entrées se trouvent dans la configuration $S = 1$ et $R = 0$ la sortie Q passe dans l'état "0". Par contre dans la configuration $S = 1$ et $R = 1$, c'est-à-dire entre t_1 et t_2 et pendant les rebonds, les sorties restent inchangées : les rebonds ne sont pas visibles sur la sortie Q. Le fonctionnement est symétrique lors du passage de la position 2 à 1.

IV.3 Bascule R.S.T ou R.S.Clock

La bascule R.S.T. est une bascule pour laquelle les entrées S et R ne sont prises en compte qu'en coïncidence avec un signal de commande. Ce signal peut être fourni par une horloge, nous avons alors une bascule synchrone. Ce circuit peut être réalisé de la façon suivante et sa représentation est donnée sur la figure 8.

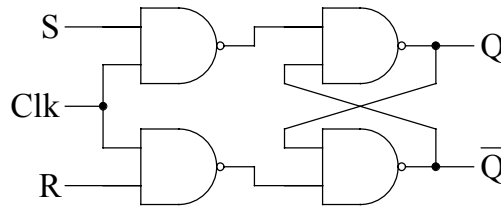


Figure 7

Lorsque le signal de commande, noté ici Clk, est à 1 la bascule fonctionne comme indiqué précédemment et les sorties suivent les variations des entrées S et R. Par contre, lorsque le signal de commande est à 0, la bascule est bloquée : Q est indépendant des éventuels changements de S et R. L'état mémorisé correspond au dernier état avant le passage de la ligne de commande de 1 à 0.

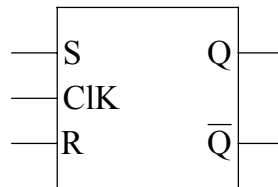


Figure 8

Dans un système synchrone le signal de commande est fourni par une horloge (clock). Celui-ci est constitué par une succession périodique d'impulsions de largeur t_p , supposée petite devant la période T. L'état de chacune des sorties restera donc bloqué pendant les intervalles séparant deux impulsions. Nous notons Q_n la valeur de la sortie Q pendant le $n^{\text{ième}}$ intervalle précédant la $n^{\text{ième}}$ impulsion et Q_{n+1} la valeur correspondante dans l'intervalle suivant la $n^{\text{ième}}$ impulsion (fig 9).

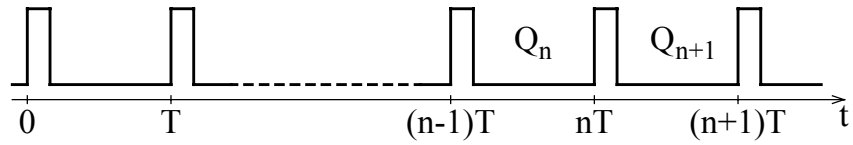


Figure 9

Considérons l'instant $t = nT + \epsilon$, où $\text{Clk} = 1$. Si $S = R = 0$, la sortie Q ne change pas donc $Q_{n+1} = Q_n$. Si $S = 1$ et $R = 0$ alors la sortie Q est forcée à 1. Si $R = 1$ et $S = 0$ alors la sortie Q est mise à 0. Si $S = R = 1$ alors les deux sorties Q et \bar{Q} sont toutes les deux à 1. Cet état est instable et ne persiste pas lorsque le signal de commande revient à 0. L'état final sera indéterminé, dépendant des vitesses relatives des portes. Cette situation ne doit pas être autorisée. L'état Q reste ensuite figé jusqu'à $(n+1)T$. Si nous notons S_n et R_n les valeurs de S et R juste avant l'instant $t = nT$ nous pouvons écrire la table de vérité :

S_n	R_n	Q_{n+1}
0	0	Q_n
1	0	1
0	1	0
1	1	?

Table 3

IV.4 Bascules J-K, D et T

IV.4.a Bascule J-K

La bascule J-K permet de lever l'ambiguïté qui existe dans la table 3. Ceci peut être obtenu en asservissant les entrées R et S aux sorties Q et \bar{Q} selon le schéma logique indiqué sur la figure 10.

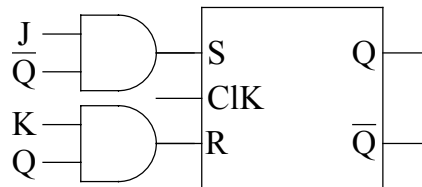


Figure 10

Nous avons alors pour les signaux R et S :

$$\begin{cases} S = J \bar{Q} \\ R = K Q \end{cases}$$

Ce qui nous permet de construire la table de vérité de la bascule J-K.

J_n	K_n	Q_n	\bar{Q}_n	S	R	Q_{n+1}
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	1	0	1	0	0	0
0	1	1	0	0	1	0
1	0	0	1	1	0	1
1	0	1	0	0	0	1
1	1	0	1	1	0	1
1	1	1	0	0	1	0

Table 4

Nous constatons que nous ne rencontrons jamais la combinaison $R = S = 1$. Cette table peut se résumer sous la forme suivante :

J_n	K_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	\bar{Q}_n

Table 5

La figure 11 explicite le diagramme logique d'une bascule J-K. Le rôle des entrées Pr et Cr sera étudié dans le prochain paragraphe.

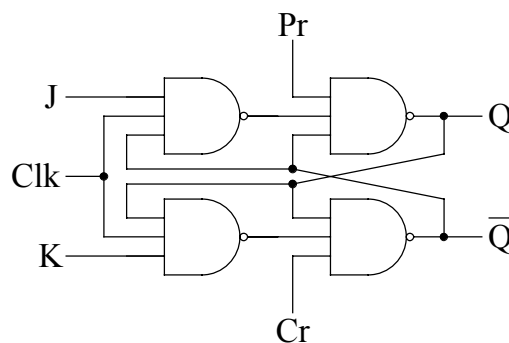


Figure 11

A partir de la table 5 nous pouvons construire la table de transition de la bascule J-K. La table 6 donne les états dans lesquels doivent se trouver les entrées J et K pour obtenir chacune des quatre transitions possibles de la sortie Q. Une croix indique que l'état de l'entrée considérée est indifférent : 0 ou 1. Par exemple, pour obtenir la transition $0 \rightarrow 1$ de la sortie Q il faut que l'entrée J soit dans l'état 1, quelque soit l'état de l'entrée K. En effet, nous pouvons avoir $J = K = 1$ qui inverse l'état de la bascule ou $J = 1$ et $K = 0$ qui charge 1 dans la bascule.

Q_n	Q_{n+1}	J_n	K_n
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table 6

Comme les deux entrées ne sont jamais spécifiées simultanément il est possible de choisir pour simplifier l'égalité des deux entrées :

$$J = K$$

On utilise parfois l'expression logique donnant Q_{n+1} en fonction de J_n , K_n et Q_n . Pour cela nous pouvons par exemple construire le tableau de Karnaugh à partir de la table de vérité (table 4) de la bascule J-K :

JK \ Q	00	01	11	10
0			1	1
1	1			1

D'où nous tirons l'équation caractéristique qui exprime l'état futur en fonction de l'état présent et des entrées :

$$Q_{n+1} = J_n \bar{Q}_n + \bar{K}_n Q_n$$

IV.4.b Preset et Clear

Les entrées asynchrones (car à utiliser en absence de signal d'horloge) Pr (Preset) et Cr (Clear) permettent d'assigner l'état initial de la bascule, par exemple à la mise sous tension pour éviter tout aléa. En fonctionnement normal ces deux entrées doivent être maintenues à 1. Lorsque le signal d'horloge est à 0 nous avons la table de vérité suivante :

Pr	Cr	Q
1	1	Q
0	1	1
1	0	0

Table 7

La figure 12 donne la représentation symbolique d'une bascule J-K avec les entrées Preset et Clear.

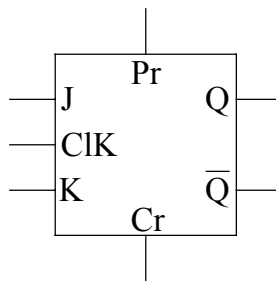


Figure 12

IV.4.c Bascule J-K Maître-Esclave

Jusqu'à présent nous avons construit les tables de vérité à partir de la logique combinatoire qui suppose que les entrées sont indépendantes des sorties. Or dans la bascule J-K nous avons introduit des connexions d'asservissement entre les entrées et les sorties. Ainsi supposons qu'avant le signal d'horloge nous avons $J = K = 1$ et $Q = 0$ (notations de la figure 11). Lorsque le signal d'horloge passe à 1 la sortie Q devient 1. Ce changement intervient après un intervalle de temps Δt . Nous avons alors $J = K = Q = 1$. D'après la table 4 nous voyons que la sortie Q doit alors revenir à 0. Ainsi la sortie Q va osciller entre 0 et 1 pendant toute la durée du signal d'horloge rendant le résultat ambigu.

Pour éviter ce problème on monte deux bascules R-S en cascade (fig. 13) en asservissant (traits épais) les entrées de la première (Maître) aux sorties de la seconde (Esclave). D'autre part, le signal d'horloge parvenant à l'esclave est inversé.

Supposons $Pr = Cr = 1$. Pendant la $n^{i\grave{e}me}$ impulsion le signal d'horloge est haut pour le maître et bas pour l'esclave. L'état Q_n est donc invariant pendant la durée t_p de l'impulsion. Le problème précédent est donc résolu et l'état de la sortie de la bascule maître Q_M est donné par la table 5. Ensuite lorsque le signal d'horloge passe à 0 pour le maître celui-ci est bloqué alors que l'esclave est libéré. Nous avons alors :

$$\begin{cases} (Q_M = 1, \bar{Q}_M = 0) \Rightarrow (S = 1, R = 0) \Rightarrow (Q_{n+1} = 1, \bar{Q}_{n+1} = 0) \\ (Q_M = 0, \bar{Q}_M = 1) \Rightarrow (S = 0, R = 1) \Rightarrow (Q_{n+1} = 0, \bar{Q}_{n+1} = 1) \end{cases}$$

L'état de la bascule maître est transféré à la bascule esclave lors de la transition $1 \rightarrow 0$ (front descendant) du signal d'horloge.

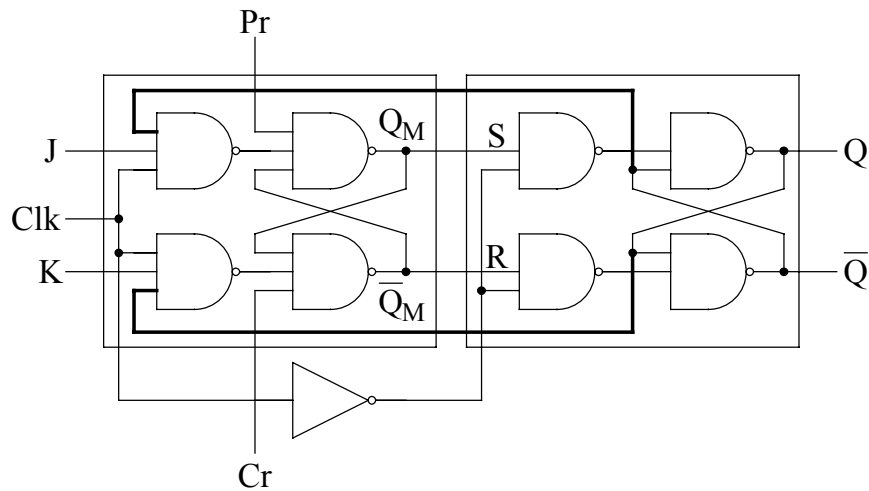


Figure 13

IV.4.d Bascule D

Une bascule D (Delay) est obtenue à partir d'une bascule J-K en envoyant simultanément une donnée sur l'entrée J et son inverse sur l'entrée K :

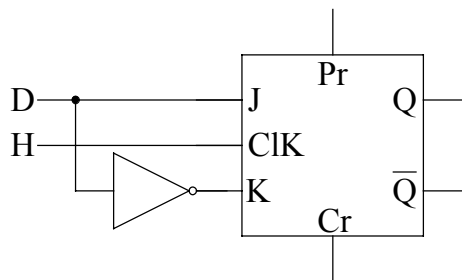


Figure 14

A partir de la table 5 nous pouvons écrire :

$$\begin{cases} D_n = 1 \Rightarrow (J_n = 1, K_n = 0) \Rightarrow Q_{n+1} = 1 \\ D_n = 0 \Rightarrow (J_n = 0, K_n = 1) \Rightarrow Q_{n+1} = 0 \end{cases}$$

Ce qui peut se résumer par $Q_{n+1} = D_n$. Ainsi l'état de la bascule Q pendant l'intervalle n+1 est égal à la valeur de l'entrée D pendant l'intervalle n. Une bascule D agit comme une unité à retard pour laquelle la sortie suit l'entrée avec un cycle de retard. Sa représentation symbolique est donnée par la figure 15.

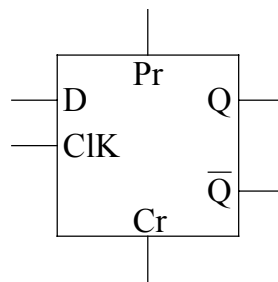


Figure 15

IV.4.e Bascule D à commande sur front montant

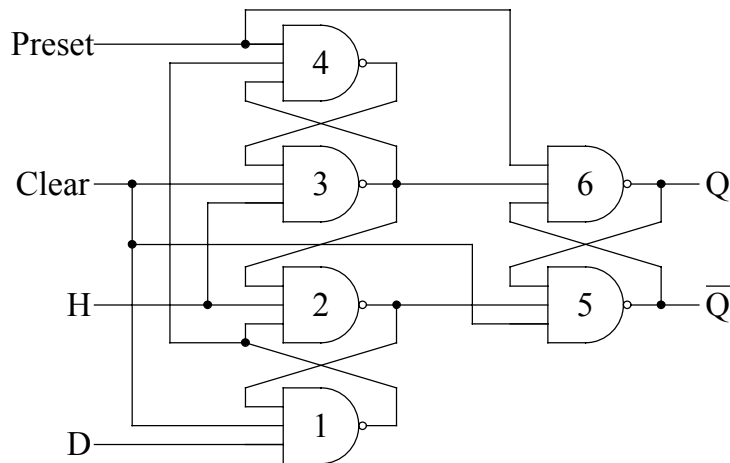


Figure 16

Dans ce deuxième type de bascule D c'est la valeur de l'entrée D présente au moment de la transition $0 \rightarrow 1$ du signal d'horloge qui est chargée dans la bascule. Ensuite, que H soit égal à 1 ou à 0, la bascule est isolée de l'extérieur jusqu'au prochain front montant du signal H. L'entrée doit être stabilisée un peu avant la transition de H et conserver cette valeur un certain temps après.

Considérons le circuit schématisé sur la figure 16, dont la représentation symbolique est donnée par la figure 17. Notons S_1, S_2, S_3 et S_4 les sorties des portes NAND numérotées respectivement 1, 2, 3 et 4. En fonctionnement normal nous devons avoir Clear = 1 pour ne pas bloquer les portes 1, 3 et 5 et Preset = 1 pour ne pas bloquer les portes 4 et 6. Pour $H = 0$ nous avons $S_2 = S_3 = 1$. Considérons maintenant le passage de H à 1.

1^{er} cas : D = 1

$$\begin{array}{llll}
 \text{Porte 1 :} & D = S_2 = \text{Clear} = 1 & \Rightarrow & S_1 = 0 \\
 \text{Porte 2 :} & S_1 = 0 & \Rightarrow & S_2 = 1 \\
 \text{Porte 4 :} & S_1 = 0 & \Rightarrow & S_4 = 1 \\
 \text{Porte 3 :} & H = S_4 = \text{Clear} = 1 & \Rightarrow & S_3 = 0 \\
 \\
 \text{Porte 6 :} & S_3 = 0 & \Rightarrow & Q = 1 \\
 \text{Porte 5 :} & Q = S_2 = \text{Clear} = 1 & \Rightarrow & \bar{Q} = 0
 \end{array}$$

2^{ème} cas : D = 0

$$\begin{array}{llll}
 \text{Porte 1 :} & D = 0 & \Rightarrow & S_1 = 1 \\
 \text{Porte 2 :} & H = S_1 = S_3 = 1 & \Rightarrow & S_2 = 0 \\
 \text{Porte 4 :} & S_1 = S_3 = \text{Preset} = 1 & \Rightarrow & S_4 = 0 \\
 \text{Porte 3 :} & S_4 = 0 & \Rightarrow & S_3 = 1 \\
 \\
 \text{Porte 5 :} & S_2 = 0 & \Rightarrow & \bar{Q} = 1 \\
 \text{Porte 6 :} & S_3 = \text{Preset} = \bar{Q} = 1 & \Rightarrow & Q = 0
 \end{array}$$

Dans les deux cas nous vérifions que $Q = D$. Maintenant considérons une transition de D alors que le signal H est encore à 1.

1^{er} cas : D = 1 \rightarrow 0 (juste avant cette transition nous avons : $S_1 = S_3 = 0, S_2 = S_4 = 1$)

$$\begin{array}{llll}
 \text{Porte 1 :} & D = 0 & \Rightarrow & S_1 = 1 \\
 \text{Porte 2 :} & S_3 = 0 & \Rightarrow & S_2 = 1 \\
 \text{Porte 4 :} & S_3 = 0 & \Rightarrow & S_4 = 1 \\
 \text{Porte 3 :} & H = S_4 = \text{Clear} = 1 & \Rightarrow & S_3 = 0
 \end{array}$$

Porte 6 : $S_3 = 0 \Rightarrow Q = 1$

Porte 5 : $Q = S_2 = \text{Clear} = 1 \Rightarrow \bar{Q} = 0$

2^{ème} cas : $D = 0 \rightarrow 1$ (juste avant cette transition nous avons : $S_1 = S_3 = 1, S_2 = S_4 = 0$)

Porte 1 : $S_2 = 0 \Rightarrow S_1 = 1$

Porte 2 : $H = S_1 = S_3 = 1 \Rightarrow S_2 = 0$

Porte 4 : $S_1 = S_3 = \text{Preset} = 1 \Rightarrow S_4 = 0$

Porte 3 : $S_4 = 0 \Rightarrow S_3 = 1$

Porte 5 : $S_2 = 0 \Rightarrow \bar{Q} = 1$

Porte 6 : $S_3 = \text{Preset} = \bar{Q} = 1 \Rightarrow Q = 0$

Dans les deux cas la sortie Q reste inchangée. Etudions maintenant les opérations de Remise à Zéro et Remise à Un :

Preset = 0 et Clear = 1 :

Preset = 0 $\Rightarrow S_4 = Q = 1$

si H = 0 alors $S_2 = 1$

si H = 1 alors $H = S_4 = \text{Clear} = 1 \Rightarrow S_3 = 0 \Rightarrow S_2 = 1$

Dans les deux cas : $Q = S_2 = \text{Clear} = 1 \Rightarrow \bar{Q} = 0$

Preset = 1 et Clear = 0 :

Clear = 0 $\Rightarrow S_1 = S_3 = \bar{Q} = 1$

$S_3 = \bar{Q} = \text{Preset} = 1 \Rightarrow Q = 0$

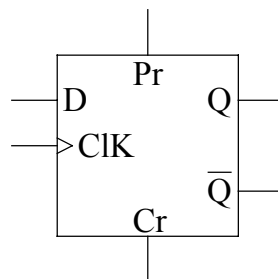


Figure 17

IV.4.f Bascule T

Dans la table 5 nous constatons que si $J = K = 1$ alors $Q_{n+1} = \overline{Q}_n$. L'état de la sortie est inversé à chaque cycle d'horloge. Une bascule T (Trigger) est obtenue à partir d'une bascule J-K en injectant le même état dans les entrées J et K (fig. 18). Sa table de vérité est donnée dans la table 7 et sa représentation par la figure 19.

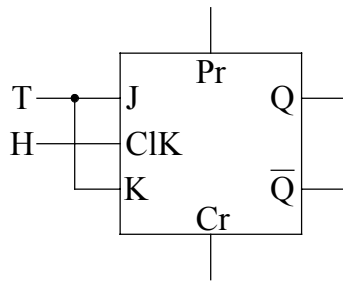


Figure 18

T_n	Q_{n+1}
1	\overline{Q}_n
0	Q_n

Table 7

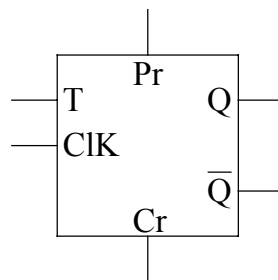


Figure 19

IV.5 Registre de mémorisation

Un registre permet la mémorisation de n bits. Il est donc constitué de n bascules, mémorisant chacune un bit. L'information est emmagasinée sur un signal de commande et ensuite conservée et disponible en lecture. La figure 20 donne un exemple de registre 4 bits réalisé avec quatre bascules D.

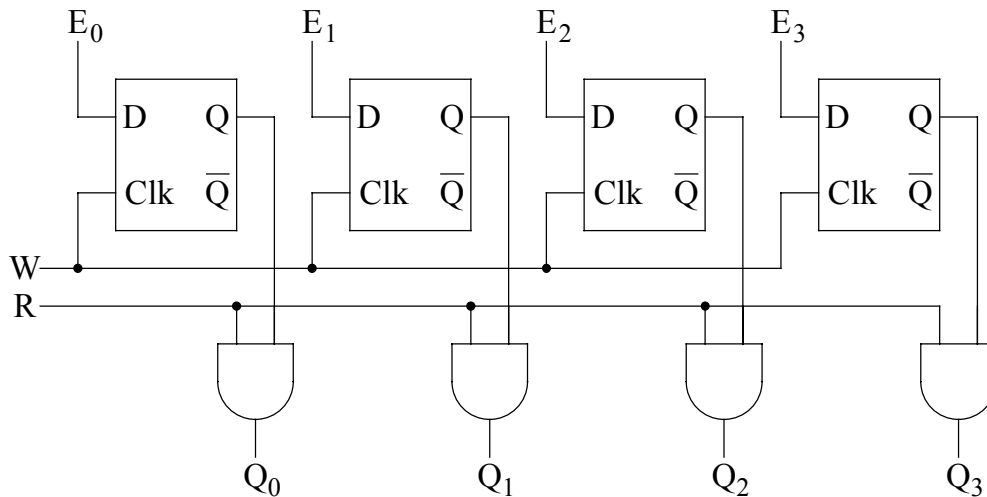


Figure 20

En synchronisme avec le signal d'écriture W le registre mémorise les états des entrées E_0 , E_1 , E_2 et E_3 . Ils sont conservés jusqu'au prochain signal de commande W. Dans cet exemple les états mémorisés peuvent être lus sur les sorties Q_0 , Q_1 , Q_2 et Q_3 en coïncidence avec un signal de validation R.

IV.6 Registre à décalage

Dans un registre à décalage les bascules sont interconnectées de façon à ce que l'état logique de la bascule de rang i puisse être transmis à la bascule de rang i+1 quand un signal d'horloge est appliqué à l'ensemble des bascules. L'information peut être chargée de deux manières dans ce type de registre.

- Entrée parallèle : comme dans le cas d'un registre de mémorisation. En général une porte d'inhibition est nécessaire pour éviter tout risque de décalage pendant le chargement parallèle.
- Entrée série : l'information est présentée séquentiellement bit après bit à l'entrée de la première bascule. A chaque signal d'horloge un nouveau bit est introduit pendant que ceux déjà mémorisés sont décalés d'un niveau dans le registre. La figure 21 schématise le chargement d'un registre 4 bits en quatre coups d'horloge.

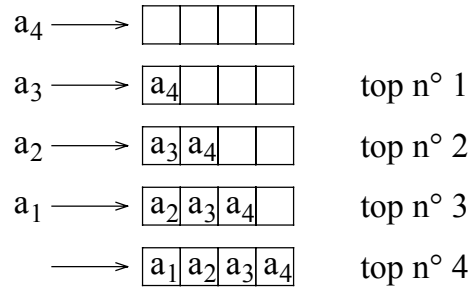


Figure 21

De même l'information peut être lue en série ou en parallèle. D'autre part, certains registres peuvent être capables de décaler à gauche et à droite. Un registre à décalage universel serait donc constitué des entrées, des sorties et des commandes suivantes :

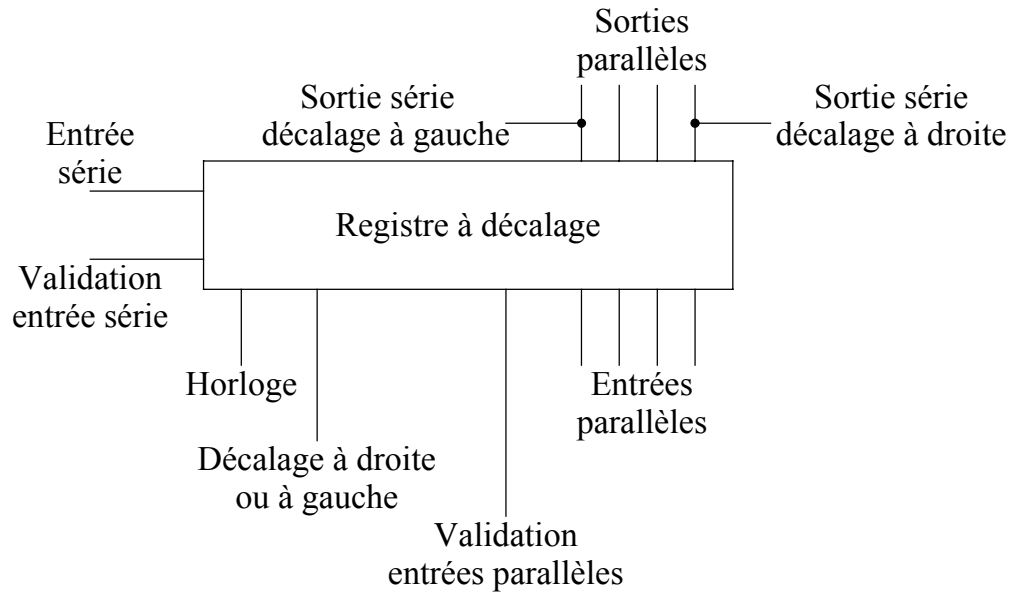


Figure 22

Généralement on utilise des bascules du type maître-esclave D ou R-S.

IV.6.a Entrée série - Sortie parallèle

La figure suivante donne un exemple de registre de 4 bits à entrée série et sortie parallèle réalisé avec des bascules D.

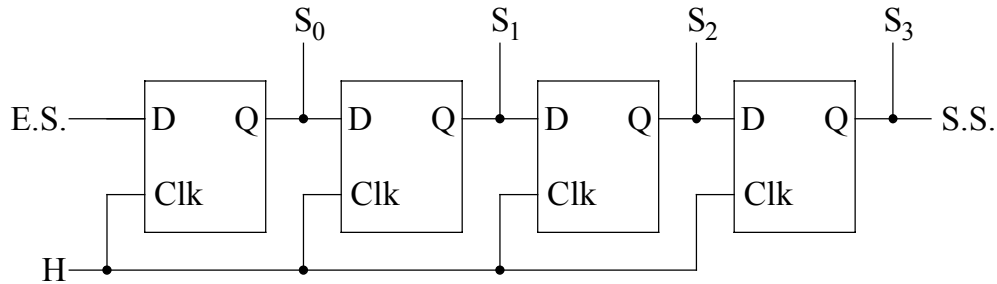


Figure 23

Ce type de registre permet de transformer un codage temporel (succession des bits dans le temps) en un codage spatial (information stockée en mémoire statique).

La sortie série peut également être utilisée. L'intérêt d'utilisation d'un registre à décalage en chargement et lecture série réside dans la possibilité d'avoir des fréquences d'horloge différentes au chargement et à la lecture. Le registre constitue alors un tampon.

IV.6.b Entrée parallèle - sortie série

La figure 24 présente un exemple de registre à décalage à entrée parallèle ou série et sortie série. Si $X = 1$ l'entrée parallèle est inhibée et l'entrée série est validée. Si $X = 0$ l'entrée série est bloquée par contre le chargement par l'entrée parallèle est autorisé.

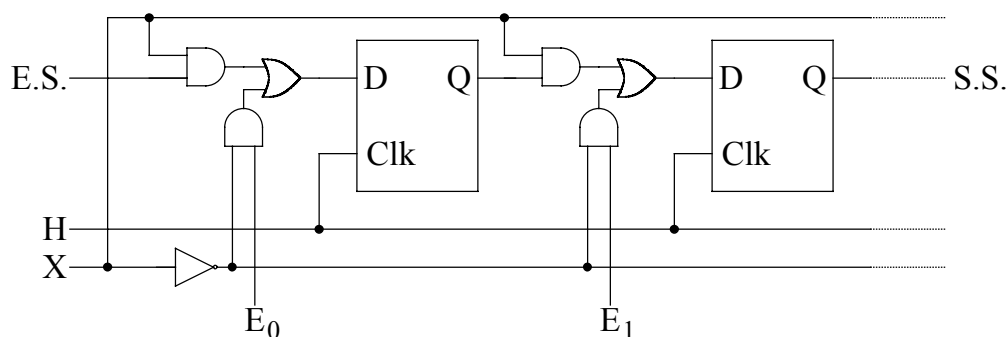


Figure 24

Un registre à décalage à entrée parallèle et sortie série transforme un codage spatial en codage temporel.

IV.6.c Entrée parallèle - Sortie parallèle

La figure suivante présente un exemple de registre à décalage avec entrées série et parallèle et sorties série et parallèle réalisé avec des bascules de type D.

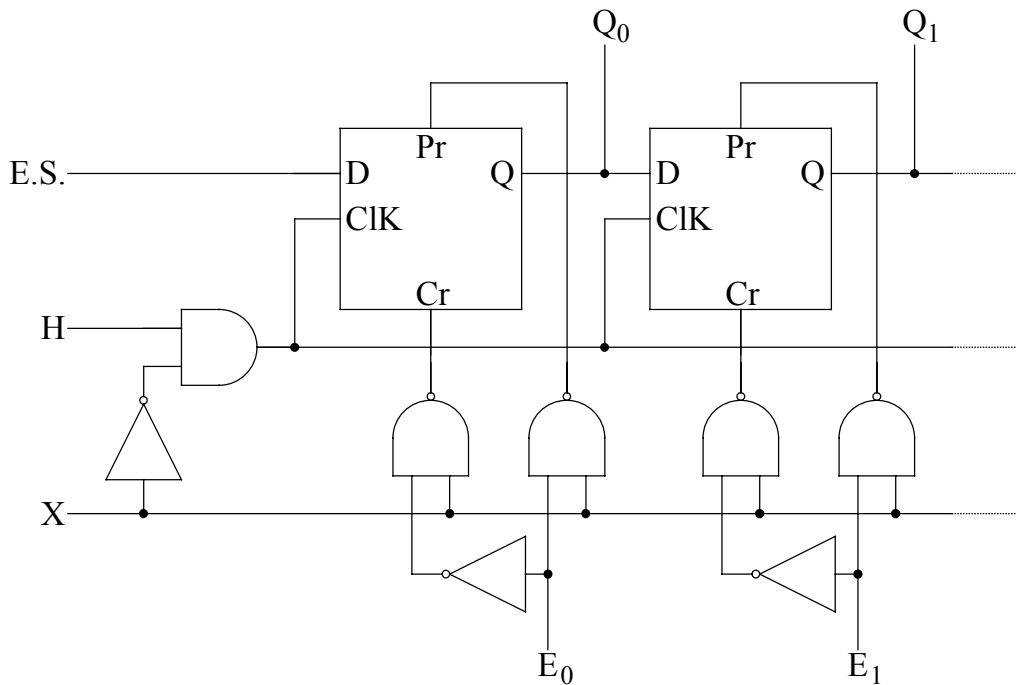


Figure 25

La commande permet de sélectionner le mode de chargement et d'inhiber le signal d'horloge en cas de chargement parallèle. Si $X = 0$ nous avons $Pr = Cr = 1$, ce qui garantit le fonctionnement normal des bascules. Si $X = 1$ alors selon l'état de chacune des entrées nous avons :

$$\left. \begin{array}{l} E_i = 1 \Rightarrow (Pr = 0, Cr = 1) \Rightarrow Q_i = 1 \\ E_i = 0 \Rightarrow (Pr = 1, Cr = 0) \Rightarrow Q_i = 0 \end{array} \right\} \Rightarrow Q_i = E_i$$

IV.6.d Registre à décalage à droite et à gauche

La figure 26 présente un exemple de registre à décalage universel de 4 bits. Les diverses possibilités sont sélectionnées par les lignes commande S_0 et S_1 . Considérons la ligne transportant le signal d'horloge aux bascules, elle est gouvernée par l'expression logique :

$$\text{Clk} = \overline{\overline{H} + \overline{S_0} \cdot \overline{S_1}} = H \cdot (S_0 + S_1)$$

Le signal d'horloge sera donc inhibé si $S_0 = S_1 = 0$.

Pour sélectionner le chargement parallèle (entrées A, B, C et D) il faut :

$$\overline{\overline{S_0} + \overline{S_1}} = S_0 \cdot S_1 = 1$$

C'est-à-dire $S_0 = S_1 = 1$. Le chargement se fera sur un signal d'horloge.

Pour sélectionner le décalage à droite (entrée E_1 , sortie Q_D) il nous faut $S_0 = 1$ et $S_1 = 0$ et pour le décalage à gauche (entrée E_0 , sortie Q_A) $S_0 = 0$ et $S_1 = 1$. Ce que nous pouvons résumer dans le tableau suivant :

S_0	S_1	Fonction
0	0	Registre bloqué
0	1	Décalage à gauche
1	0	Décalage à droite
1	1	Chargement parallèle

Table 8

Un registre à décalage à droite et à gauche permet d'effectuer des multiplications et des divisions entières par des puissances de 2. En effet une multiplication par 2 est équivalente à un décalage vers la gauche et une division par 2 à un décalage vers la droite. Une multiplication par 2^n sera obtenue par n décalages à gauche et une division par 2^n par n décalages à droite.

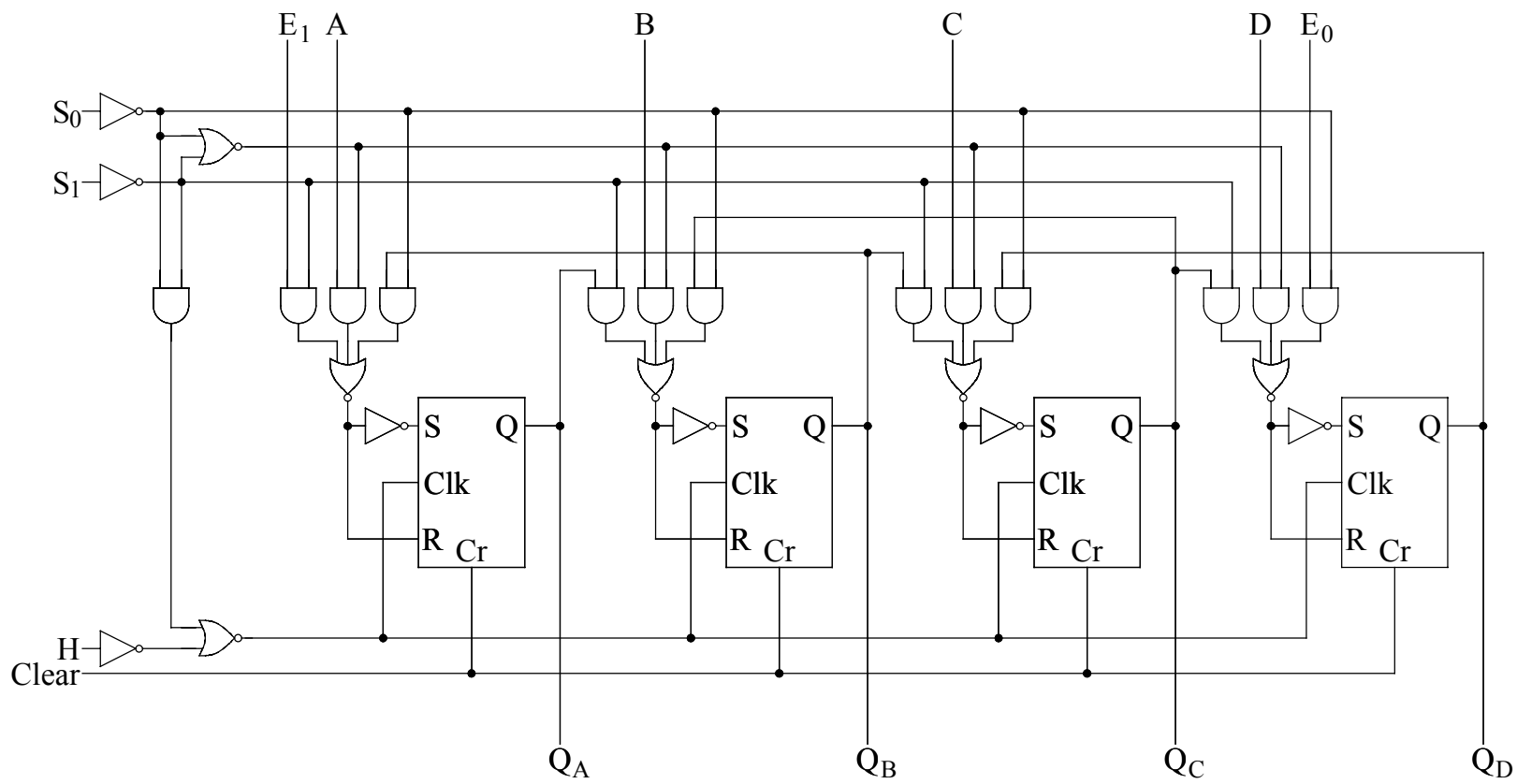


Figure 26

IV.7 Compteurs

Un compteur est un ensemble de n bascules interconnectées par des portes logiques. Ils peuvent donc mémoriser des mots de n bits. Au rythme d'une horloge ils peuvent décrire une séquence déterminée c'est-à-dire occuper une suite d'états binaires. Il ne peut y avoir au maximum que 2^n combinaisons. Ces états restent stables et accessibles entre les impulsions d'horloge. Le nombre total N des combinaisons successives est appelé le modulo du compteur. On a $N \leq 2^n$. Si $N < 2^n$ un certain nombre d'états ne sont jamais utilisés.

Les compteurs binaires peuvent être classés en deux catégories :

- les compteurs asynchrones;
- les compteurs synchrones.

De plus on distingue les compteurs réversibles ou compteurs-décompteurs.

IV.8 Compteurs asynchrones

Un compteur asynchrone est constitué de n bascules J-K fonctionnant en mode T. Le signal d'horloge n'est reçu que par le premier étage (bascule LSB : Least Significant Bit). Pour chacune des autres bascules le signal d'horloge est fourni par une sortie de la bascule de rang immédiatement inférieur.

Considérons par exemple (fig. 27) un compteur modulo 8 suivant le code binaire pur constitué de trois bascules J-K maîtres-esclaves.

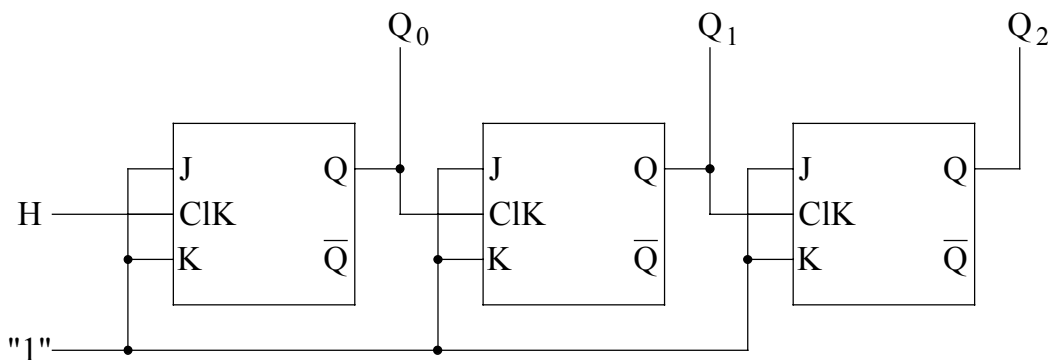


Figure 27

Supposons les trois bascules à zéro à l'instant $t = 0$. Nous avons vu que pour une bascule maître-esclave la sortie change d'état juste après le passage du signal d'horloge de l'état 1 à l'état 0 (front descendant). L'évolution temporelle des trois sorties Q_0 , Q_1 et Q_2 par rapport aux impulsions d'horloge est représentée sur la figure 28. La sortie Q_0 bascule sur chaque front descendant du signal d'horloge. La sortie Q_1 change d'état à chaque transition $1 \rightarrow 0$ de la sortie

Q_0 . De même le basculement de la sortie Q_2 est déclenché par une transition $1 \rightarrow 0$ de la sortie Q_1 .

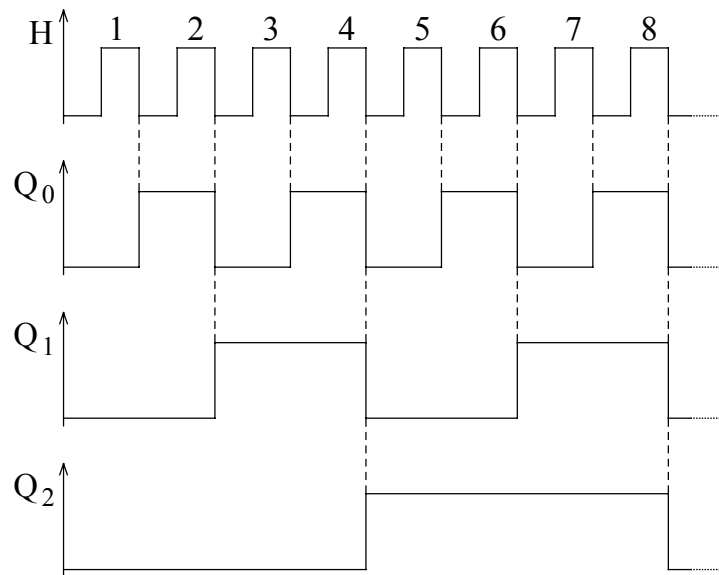


Figure 28

A partir de ce chronogramme nous pouvons écrire la liste des états successifs des trois sorties :

Impulsion	Q_2	Q_1	Q_0
état initial	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

Table 9

Nous avons réalisé un compteur s'incrémentant d'une unité à chaque top d'horloge, avec un cycle de huit valeurs de 0 à 7 (modulo 8).

Nous constatons que les sorties Q_0 , Q_1 et Q_2 fournissent des signaux périodiques de fréquences respectivement 2, 4 et 8 plus faibles. La division de fréquence est une des applications des compteurs.

IV.8.a Compteur-décompteur asynchrone

Nous obtenons un compteur en déclenchant chaque bascule lorsque celle de rang immédiatement inférieur passe de l'état 1 à 0. Pour réaliser un décompteur il faut que le changement d'état d'une bascule intervienne lorsque la bascule de rang immédiatement inférieur passe de l'état 0 à 1. Pour cela il suffit d'utiliser la sortie \bar{Q} de chaque bascule pour déclencher la suivante.

On réalise un compteur-décompteur en utilisant un multiplexeur 2 entrées - 1 sortie entre chaque étage pour sélectionner la sortie à utiliser. Pour l'exemple présenté sur la figure 29, selon l'état de la ligne de commande X nous pouvons sélectionner le mode de comptage :

- X = 1 \Rightarrow compteur;
- X = 0 \Rightarrow décompteur.

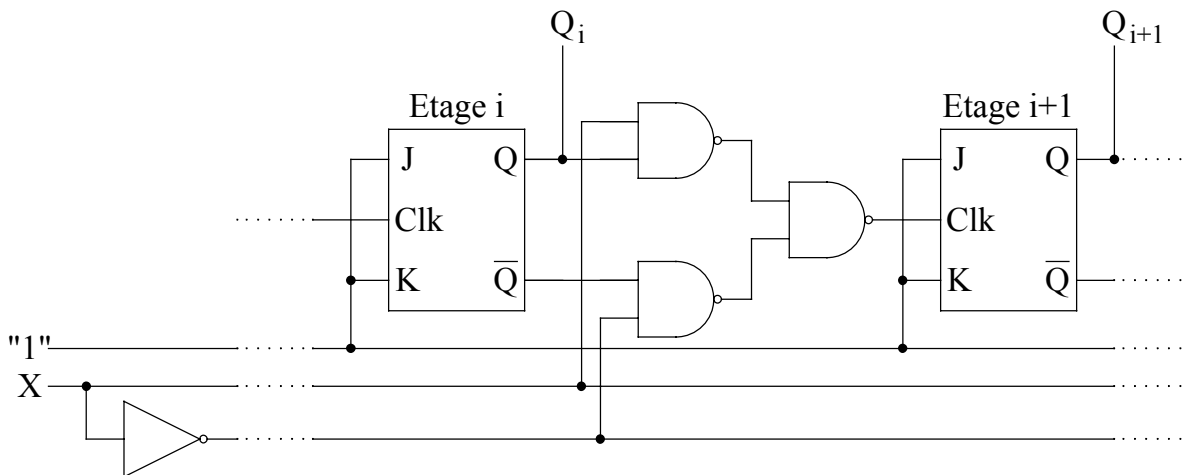


Figure 29

IV.8.b Remise à Zéro et chargement d'un compteur

La figure 30 présente un exemple de montage permettant de remettre à zéro un compteur ou de le charger avec une valeur déterminée. Pour cela on utilise les entrées asynchrones des bascules. En fonctionnement normal du compteur nous devons avoir : DS = R = 1. Nous avons alors : J = K = Pr = Cr = 1 sur chaque bascule du compteur.

Pour RAZ : R = 0

- \Rightarrow J = K = 0 Interdit tout basculement sur une impulsion du signal Clk;
- \Rightarrow (Pr = 1, Cr = 0) \Rightarrow Q = 0.

Chargement : (DS = 0, R = 1)

$\Rightarrow J = K = 0$ Interdit tout basculement sur une impulsion du signal Clk;

$D = 0 \Rightarrow (Pr = 1, Cr = 0) \Rightarrow Q = 0$

$D = 1 \Rightarrow (Pr = 0, Cr = 1) \Rightarrow Q = 1$

Dans ces deux cas nous obtenons $Q = D$. Nous sommes donc capable de charger chaque bit du compteur avec une valeur donnée à présenter sur l'entrée D, donc d'initialiser le compteur.

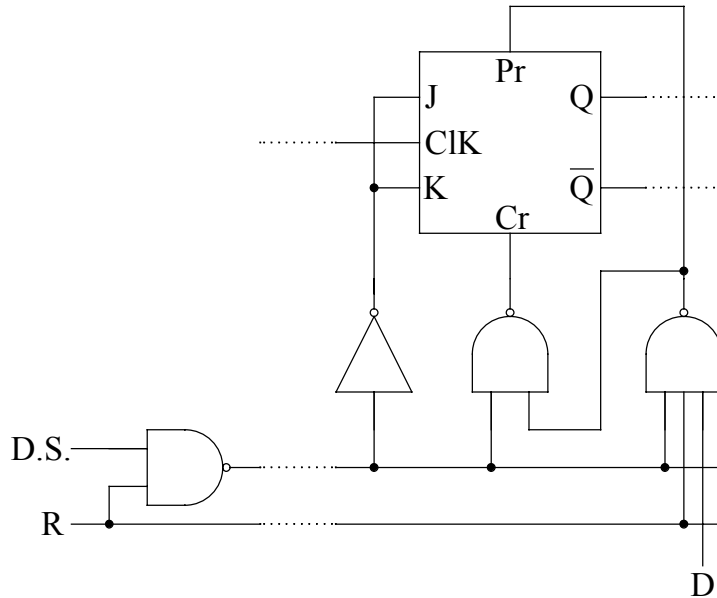


Figure 30

IV.8.c Compteur à cycle incomplet

On peut souhaiter compter jusqu'à un nombre N qui ne soit pas une puissance de 2, par exemple 10 (système décimal). Pour cela on utilise un compteur de n bascules, tel que $2^n > N$. On lui ajoute un asservissement de l'entrée Clear pour remettre le compteur à zéro tous les N coups.

Considérons par exemple un compteur modulo 10. Nous voulons que l'entrée Clear soit à 0 lorsque le compteur atteint $10_{10} = 1010_2$. Pour cela nous pouvons écrire l'expression logique :

$$\overline{Cr} = Q_3 \cdot \overline{Q_2} \cdot Q_1 \cdot \overline{Q_0}$$

En fait dans ce cas particulier nous pouvons simplifier cette relation logique en ne tenant compte que des sorties à 1 dans l'expression binaire de N. En effet il ne peut y avoir ambiguïté : toute combinaison contenant les mêmes sorties à 1 et au moins une autre à 1 correspond à un nombre plus grand que N et ne peut être rencontrée dans la séquence décrite par le compteur. Pour un compteur modulo 10 nous pouvons donc utiliser :

$$\overline{Cr} = Q_3 \cdot Q_1$$

ce qui nous conduit au schéma suivant :

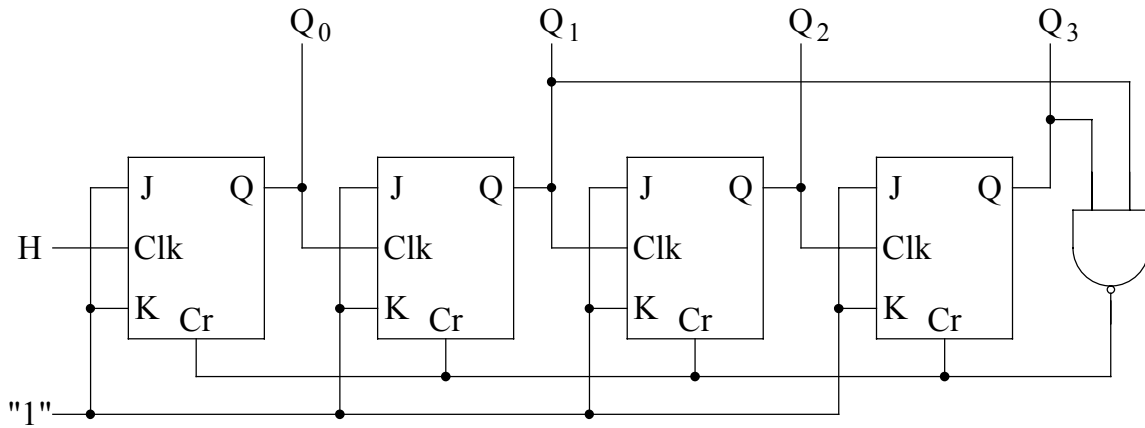


Figure 31

IV.8.d Inconvénients des compteurs asynchrones

Comme chaque bascule a un temps de réponse le signal d'horloge ne parvient pas simultanément sur toutes les bascules. Ceci a pour conséquence de provoquer des états transitoires qui peuvent être indésirables. Supposons le même temps de réponse t_r pour toutes les bascules. Considérons la chronologie du passage d'un compteur asynchrone 4 bits de 0111 à 1000. Celle-ci est présentée sur la figure 32. Nous constatons que le compteur passe par les états transitoires 0110, 0100 et 0000 qui sont faux. Ceci est un inconvénient rédhibitoire chaque fois que le compteur est exploité par des organes rapides.

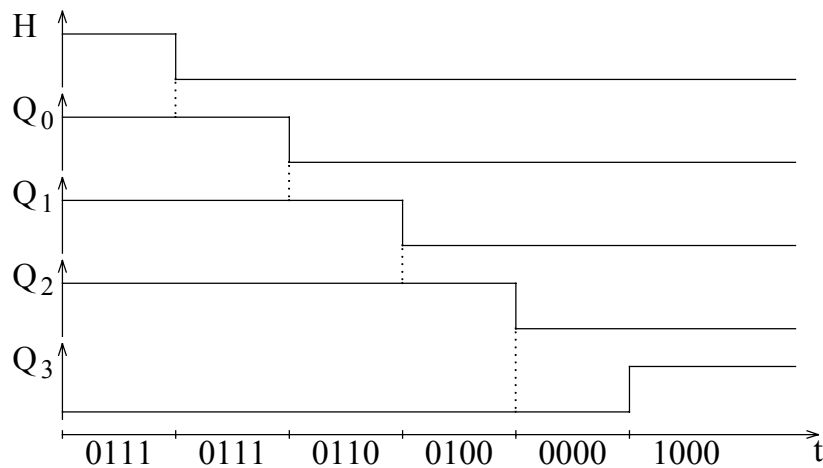


Figure 32

IV.9 Compteurs synchrones

Dans un compteur synchrone toutes les bascules reçoivent en parallèle le même signal d'horloge. Pour faire décrire au compteur une séquence déterminée il faut à chaque impulsion d'horloge définir les entrées synchrones J et K. Pour cela on utilise la table de transition de la bascule J-K (table 6). Nous avons déjà remarqué que cette table peut se simplifier. En effet, pour chacune des quatre transitions possibles une seule des entrées J ou K est définie. Rien ne nous interdit donc de les mettre dans le même état, c'est-à-dire $J = K$, comme dans une bascule T.

Prenons l'exemple d'un compteur synchrone 3 bits fonctionnant selon le code binaire pur. Nous pouvons dresser un tableau précisant les valeurs des entrées J et K permettant d'obtenir chaque transition (passage d'une ligne à la suivante). Pour qu'une bascule change d'état il faut que ses deux entrées soient à 1.

# top	Q_2	Q_1	Q_0	$J_2 = K_2$	$J_1 = K_1$	$J_0 = K_0$
0	0	0	0	0	0	1
1	0	0	1	0	1	1
2	0	1	0	0	0	1
3	0	1	1	1	1	1
4	1	0	0	0	0	1
5	1	0	1	0	1	1
6	1	1	0	0	0	1
7	1	1	1	1	1	1
8	0	0	0			

Table 10

Chaque ligne de cette table correspond à une même tranche de temps. Il est assez facile d'en déduire les expressions logiques reliant les entrées aux sorties :

$$\begin{cases} J_0 = K_0 = 1 \\ J_1 = K_1 = Q_0 \\ J_2 = K_2 = Q_0 \cdot Q_1 \end{cases}$$

De manière générale nous pouvons vérifier que les équations de commutation satisfont les relations de récurrence suivantes :

$$\begin{cases} J_0 = K_0 = 1 \\ J_i = K_i = Q_0 \cdot Q_1 \cdot \dots \cdot Q_{i-1} \end{cases}$$

ou encore :

$$\begin{cases} J_0 = K_0 = 1 \\ J_i = K_i = J_{i-1} \cdot Q_{i-1} \end{cases}$$

Procédons de même pour réaliser un décompteur, nous écrivons la table des transitions recherchées :

# top	Q ₂	Q ₁	Q ₀	J ₂ = K ₂	J ₁ = K ₁	J ₀ = K ₀
0	1	1	1	1	1	1
1	1	1	0	0	0	1
2	1	0	1	0	1	1
3	1	0	0	0	0	1
4	0	1	1	1	1	1
5	0	1	0	0	0	1
6	0	0	1	0	1	1
7	0	0	0	0	0	1
8	0	0	0			

Table 11

Nous en déduisons l'expression logique des entrées d'un décompteur :

$$\begin{cases} J_0 = K_0 = 1 \\ J_1 = K_1 = \overline{Q_0} \\ J_2 = K_2 = \overline{Q_0} \cdot \overline{Q_1} \end{cases}$$

Nous constatons que les équations de commutation sont identiques en utilisant cette fois les sorties complémentaires \overline{Q} .

Aux deux manières d'exprimer les relations de récurrence des équations de commutation correspondent deux types de circuits. Le premier (fig. 33) est dit à report parallèle, le second (fig. 34) à report série. Dans le report série on utilise la fonction J_{i-1} . On évite ainsi des portes à multiples entrées. Par contre, il faut tenir compte du retard dans l'établissement de J_{i-1} . Il faut donc que la largeur des impulsions d'horloge soit assez grande et la vitesse maximum de fonctionnement sera plus faible que pour le report parallèle.

Les deux schémas présentés sur les figures 33 et 34 correspondent à des compteurs-décompteurs ($X = 0 \Rightarrow$ compteur, $X = 1 \Rightarrow$ décompteur).

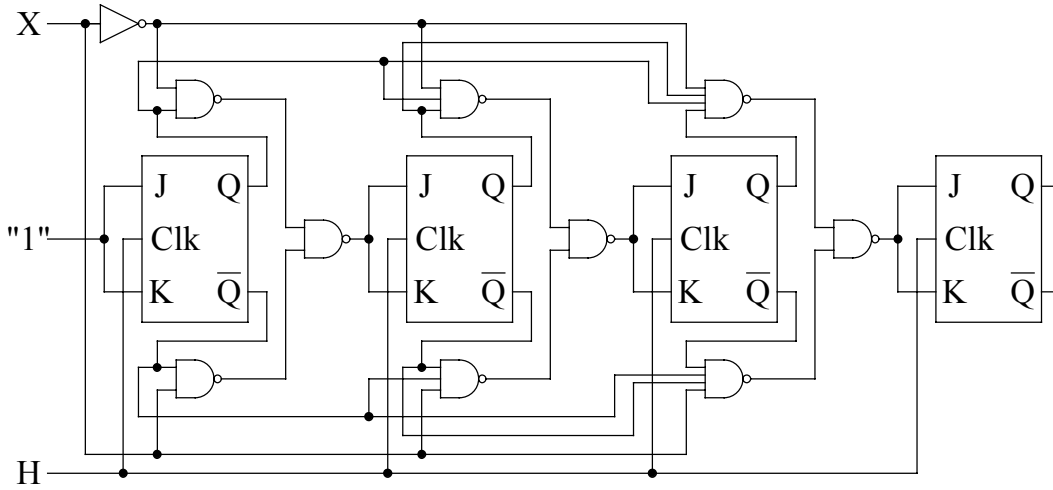


Figure 33

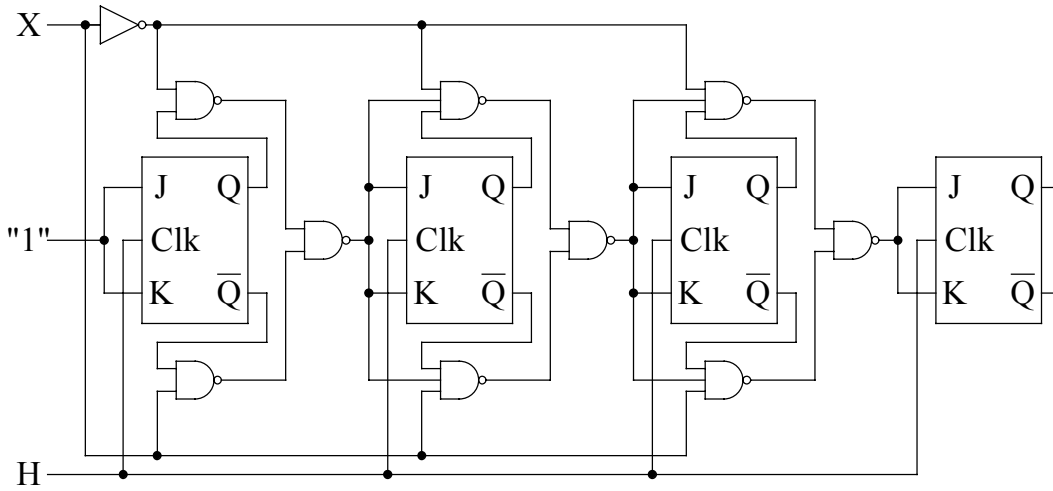


Figure 34