

## Chapitre 8

## Circuits séquentiels élémentaires

Dans un circuit combinatoire, les valeurs des sorties à un instant donné sont directement imposées par celles des entrées. Ce type de circuits ne permet en fait de traiter qu'une classe restreinte de problèmes ne nécessitant pas de besoin de mémorisation. Les circuits séquentiels ont quant à eux la capacité de mémoriser des informations et par conséquent de traiter des séquences de données. L'étude des circuits séquentiels élémentaires (bascules, registres, mémoires, compteurs) fera l'objet de ce chapitre.

## 8.1. Bascules

## 8.1.1. Notions de circuit séquentiel et de point mémoire

A l'inverse des circuits combinatoires, l'apparition d'un même vecteurs d'entrée sur un circuit séquentiel n'entraîne pas nécessairement des valeurs identiques sur les sorties. A titre d'exemple, considérons le circuit présenté sur la figure 8.1. Les combinaisons (01, 00) et (11, 00) appliquées sur les entrées conduisent à une valeur de sortie différente sur la sortie et ceci, malgré le fait que le dernier vecteur soit identique (00).

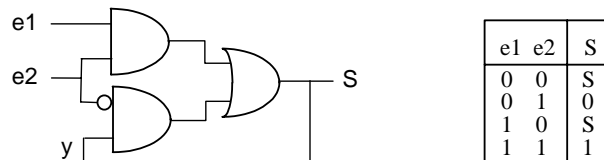


Figure 8.1. Circuit séquentiel

De manière générale, les circuits séquentiels font apparaître des **boucles de rétroaction** qui permettent de mémoriser des informations relatives aux stimuli antérieurs appliqués sur le circuit. La sortie d'un circuit séquentiel est ainsi fonction de variables internes ( $y$ ) également appelées variables d'état.

Un système séquentiel peut être synchrone ou asynchrone selon qu'il est commandée ou pas par un signal d'horloge. En d'autre terme, un système séquentiel est *asynchrone* si à partir de l'instant où on applique un vecteur d'entrée, son évolution est incontrôlable de l'extérieur. Il est *synchrone* si son évolution est contrôlable de l'extérieur par un signal d'horloge.

**Les bascules** sont les circuits séquentiels élémentaires permettant de mémoriser une information binaire (bit) sur leur sortie. Elles constituent le **point mémoire** élémentaire. Elles peuvent être synchrones ou asynchrones mais toutes ont au minimum trois modes de fonctionnement (et par conséquent au moins 2 commandes):

positionnement de la sortie à 0, positionnement de la sortie à 1 et mémorisation de l'information portée par la sortie.

### 8.1.2. Bascule RS

La bascule RS est le dispositif de mémorisation élémentaire. Cette bascule est asynchrone. Toutes les bascules, y compris les bascules synchrones, ne sont en fait que des évolutions de cette bascule.

La bascule RS est un dispositif à deux entrées R (pour Reset) et S (pour Set) et une sortie Q présentant la propriété suivante :

- lorsque S et R sont à 0, Q conserve sa valeur (Etat mémoire)
- une apparition (même fugitive) de S entraîne durablement Q=1
- une apparition (même fugitive) de R entraîne durablement Q=0

Appelons  $Q_n$ , la sortie à l'instant n et  $Q_{n+1}$  la sortie à l'instant n+1. Nous pouvons dresser la table de vérité et la table de Karnaugh définissant  $Q_{n+1}$ .

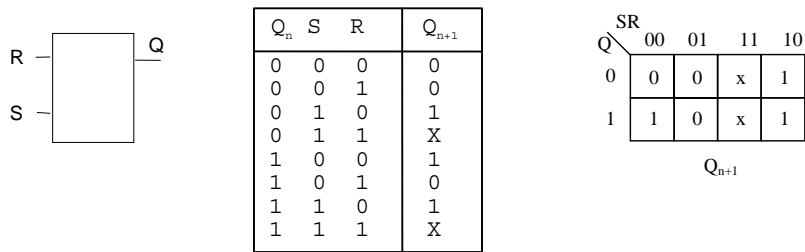


Figure 8.2. Définition de  $Q_{n+1}$

On s'aperçoit que l'énoncé du problème est incomplet : les combinaisons telles que SR=11 (3 et 7) ne sont pas définies. Elles correspondent à des ordres d'enclenchement (SET) et de déclenchement (RESET) simultanés. En laissant le problème incomplètement spécifié, on peut obtenir plusieurs équations de la bascule.

$$Q_{n+1} = S \cdot R' + Q_n \cdot R' = (S + Q_n) R' = ((S + Q_n)' + R)' \quad (1)$$

$$Q_{n+1} = S + Q_n \cdot R' = (S' \cdot (Q_n \cdot R))' \quad (2)$$

Les schémas correspondant à ces équations sont donnés sur la Figure 8.3.

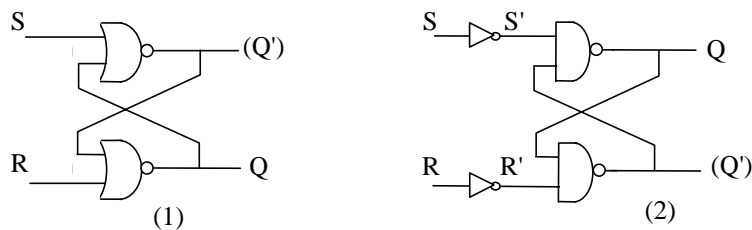


Figure 8.3. Bascules RS

Dans le cas (1) les valeurs indéterminées sont affectées à 0. La bascule est dite à déclenchement prioritaire. Dans le cas (2) les valeurs indéterminées sont affectées à 1. La bascule est dite à enclenchement prioritaire.

Sur ce type de bascule, la combinaison (R,S)=(1,1) doit être interdite car elle peut introduire une indétermination. En effet, le passage de la combinaison (R,S)=(1,1) à (R,S)=(0,0) entraîne deux valeurs possibles sur Q selon que R ou S commute en premier. Si l'on interdit la combinaison (R,S)=(1,1) on remarque

que sur les deux structures, la connexion symétrique de la sortie Q porte la valeur Q'. La table de vérité de la bascule RS est représentée sur la Figure 8.4.

S	R	$Q_{n+1}$
0	0	$Q_n$
0	1	0
1	0	1
1	1	Interdit

Figure 8.4. Table de vérité de la bascule RS

L'avantage de cette bascule est sa simplicité. Ces inconvénients sont le fait qu'elle soit purement asynchrone, qu'elle soit sensible aux parasites (tout événement sur une des entrées affecte la sortie), et qu'il existe un état interdit.

Cette bascule RS est toutefois utilisée dans un certain nombre de procédés tels que les systèmes anti-rebond. La Figure 8.5 montre que la fermeture d'un bouton poussoir peut être soumise à des rebonds et qui engendrer des commutations parasites indésirables.

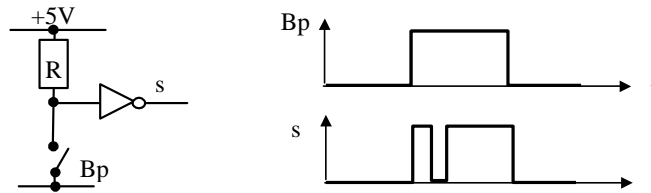


Figure 8.5. Bouton poussoir sans système anti-rebond

La Figure 8.5 montre comment à l'aide d'une bascule RS, ces commutations parasites peuvent être éliminées. L'état mémoire permet en effet de filtrer ces transitions.

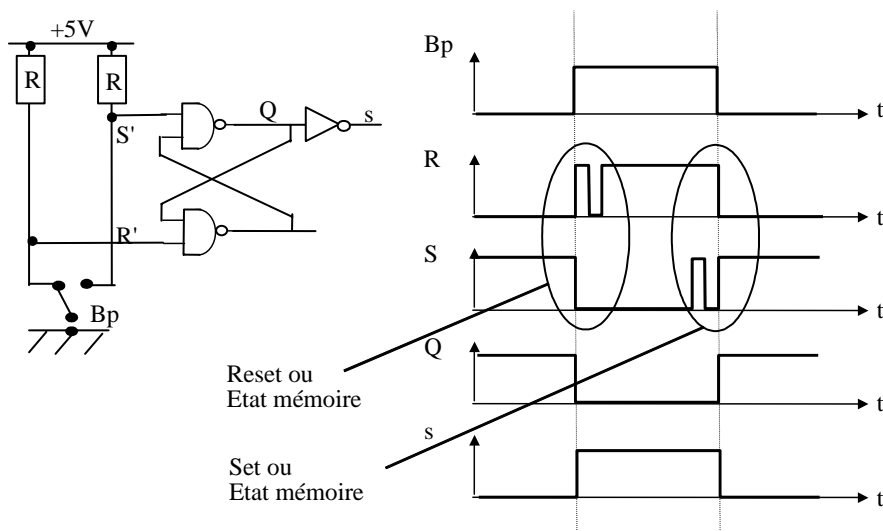


Figure 8.6. Dispositif anti-rebond

### 8.1.3. Bascule RSH

La bascule RSH est une bascule RS synchronisée par un signal d'horloge H. Lorsque H=0, la bascule est dans l'état mémoire. Lorsque H=1, la bascule fonctionne comme une bascule RS. Le schéma de cette bascule est donné sur la figure 8.7.

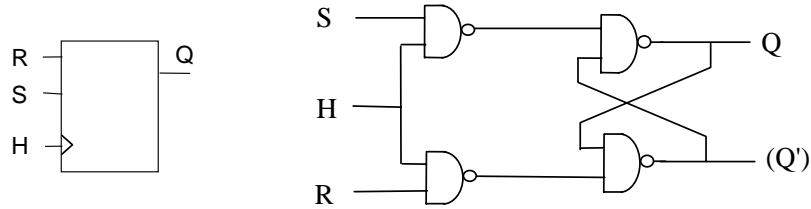


Figure 8.7. Bascule RSH

Cette bascule a toujours un état interdit et fonctionne sur les niveaux d'horloge. Tout en restant sensible aux parasites elle l'est moins que la bascule RS puisqu'elle est uniquement sensible sur le niveau haut de l'horloge (plus le niveau haut de l'horloge est réduit, moins cette bascule est sensible aux parasites).

### 8.1.4. Bascule D-latch

La bascule D-Latch est une bascule ayant une entrée D et une sortie Q synchronisée par un signal d'horloge H. Sa table de vérité est la suivante (Figure 8.8).

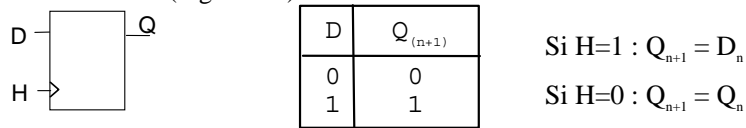


Figure 8.8. Table de vérité de la D-latch

La bascule D-Latch est une bascule qui peut être conçue sur le même principe que la RSH. Dans ce cas, elle est obtenue à partir d'une bascule RSH en ne considérant que les deux combinaisons (R,S) = (0,1) et (1,0). Cette bascule n'a donc pas d'état interdit. Elle est transparente sur le niveau haut de l'horloge (Q=D) et mémorise la valeur de sortie sur le niveau bas. Ce dispositif est en fait l'élément mémoire de base.

La structure d'un tel dispositif peut être obtenue à partir de la structure de la bascule RSH mais également directement en repartant du cahier des charges. Deux schémas à portes de cette bascule sont donnés sur la Figure 8.9.

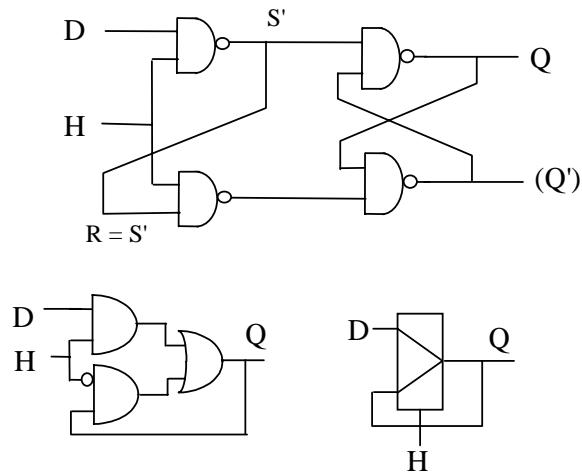


Figure 8.9. Bascule D-Latch

**Remarque :** Bien qu'il ne soit pas question ici d'implantation technologique, nous pouvons souligner que l'intérêt d'un tel dispositif, outre le fait qu'il constitue l'élément mémoire de base, réside dans la compacité de son implantation en logique 3 états et notamment en technologie CMOS. La figure 8.10 présente la structure symbolique d'une telle bascule dans ces technologies.

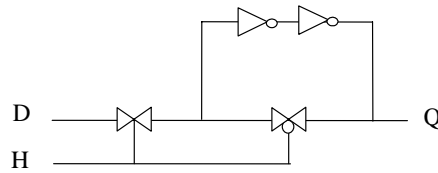


Figure 8.10. Bascule D-Latch en logique 3 états

### 8.1.5. Bascule D

La bascule D est la cellule mémoire fondamentale utilisée dans la grande majorité des applications. Ce dispositif fonctionne sur un front d'horloge (front montant ou front descendant). La table de vérité d'une bascule D est la suivante (Figure 8.11).

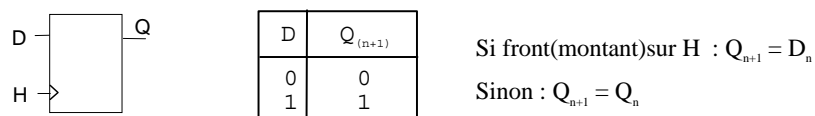


Figure 8.11. Table de vérité de la bascule D

Cette bascule peut être réalisée en cascader 2 bascules D-Latch (dispositif maître esclave) comme indiqué sur la Figure 8.12.

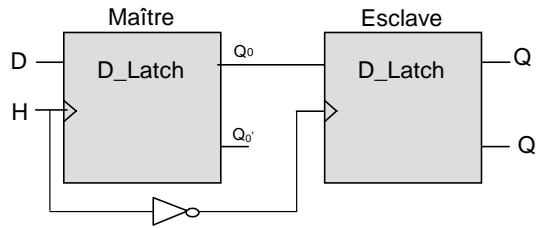


Figure 8.12. Bascule D

Un des gros avantages de ce dispositif est sont immunité aux transitoires ou parasites. En effet, la mise en cascade de 2 D-latch fonctionnant sur des niveaux opposés de l'horloge permet d'éviter la propagation de signaux transitoires à travers la bascule.

### 8.1.6. Bascule T

La bascule T (T pour Toggle) est un élément qui interprète son entrée de commande T, non comme une entrée à mémoriser, mais comme un ordre de changement d'état. Cette bascule est particulièrement intéressante à utiliser pour certaines applications et notamment pour la réalisation de compteurs. Sa table de vérité est la suivante (Figure 8.13).

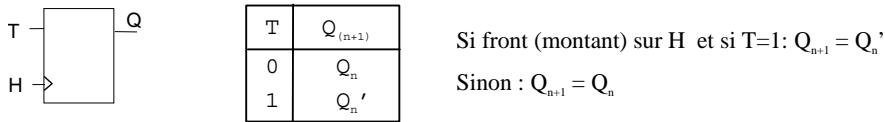


Figure 8.13. Table de vérité de la bascule T

Une telle bascule peut être réalisée à partir d'une bascule D avec :

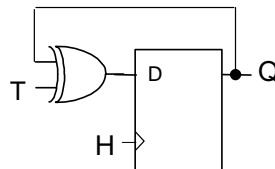
$$D_n = T' \cdot Q_n + T \cdot Q_n'$$


Figure 8.14. Bascule T réalisée avec une bascule D

### 8.1.7. Bascule JK

La bascule JK est une bascule fonctionnement sur front d'horloge (maître-esclave) comportant 2 entrées (J et K). La table de vérité de cette bascule est donnée sur la Figure 8.15.

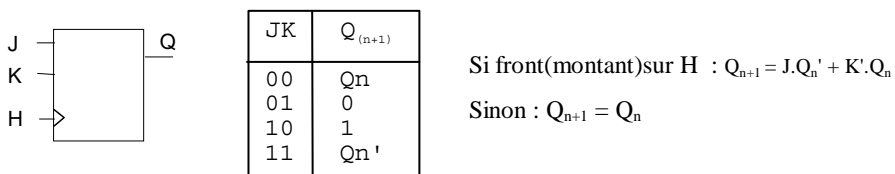


Figure 8.15. Table de vérité de la bascule JK

Comme la bascule T, cette bascule peut être réalisée à partir d'une bascule D de la manière suivante (Figure 8.16) :

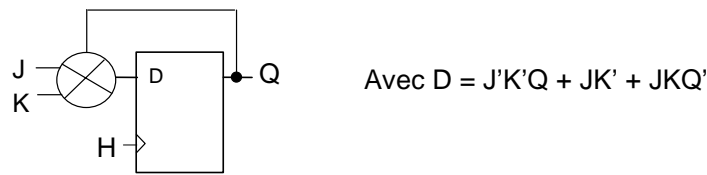


Figure 8.16. Bascule JK réalisée avec une bascule D

### 8.1.8. Initialisation des bascules

Une bascule, quel que soit son type, peut être initialisée par l'intermédiaire d'une combinaison de ses entrées et d'un coup d'horloge. Sur la plupart des bascules, il existe également des signaux spécifiques d'initialisation asynchrone, c'est à dire ne nécessitant pas de coup d'horloge. Ces entrées d'initialisation sont généralement appelées « Clear » pour la remise à 0 de la sortie et « Preset » pour la remise à 1. Elle agissent sur l'étage esclave des bascules. Un exemple de bascule avec entrées d'initialisation asynchrone « Clear » et « Preset » est donné sur la Figure 8.17.

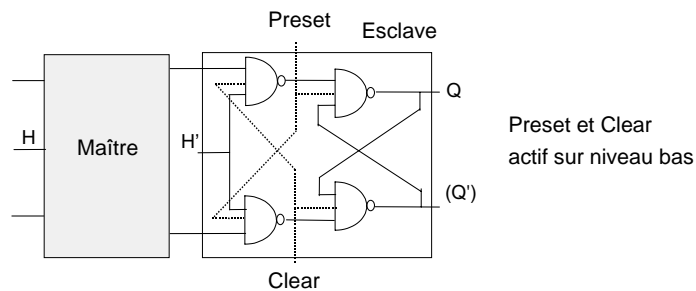


Figure 8.17. Bascule avec Clear et Preset

L'initialisation d'une bascule peut également être envisagée de manière synchrone. Pour cela il faut prévoir un dispositif intervenant directement sur les entrées synchrones des bascules comme présenté sur la figure 8.18 dans le cas d'une bascule D.

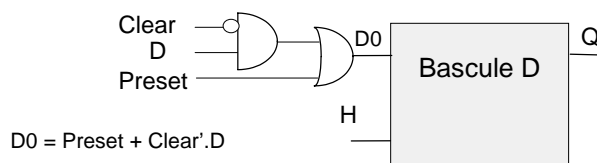


Figure 8.18. Initialisation synchrone d'une bascule D

### 8.1.9. Inhibition du fonctionnement des bascules

Pour certaines applications utilisant des bascules, le fonctionnement des bascules doit pouvoir être inhibé (conservation de l'état quoi qu'il arrive notamment sur l'horloge) par un signal d'inhibition (Inib). Pour cela plusieurs solutions peuvent être envisagées mais certaines sont à proscrire absolument sous peine de sérieux ennuis.

Par exemple, le blocage, par une porte, du signal d'horloge pour maintenir l'état d'une bascule, est une erreur fondamentale que l'on rencontre parfois. Cette approche provoque d'une part, des décalages temporels entre les signaux d'horloge (*clock skew*) qui peuvent conduire à comportements imprévisibles des bascules concernés et d'autre part, des possibilités d'impulsions parasites sur les signaux d'horloge qui peuvent également conduire à des fonctionnement imprévisibles.

Ainsi, la seule solution viable pour inhiber le fonctionnement d'une bascule est d'agir sur les entrées synchrones des bascules comme présenté sur la figure 8.19 dans le cas d'une bascule D.

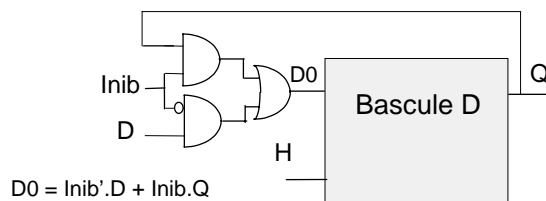


Figure 8.19. Inhibition du fonctionnement d'une bascule D

### 8.1.10. Paramètres temporels des bascules

Pour qu'une bascule fonctionne correctement, il est nécessaire que le signal présent sur les entrées de la bascule (D ou JK) soit stabilisé depuis un certain temps lorsque le front d'horloge actif intervient (temps de « setup ») et reste stable pendant un certain temps après ce front d'horloge (temps de « hold » ou de maintien).

D'autre part, la commutation des sorties d'une bascule se fait avec un certain temps de retard par rapport au signal qui a produit cette commutation (Horloge, Reset ou Preset). Ces retards peuvent être différents selon le signal qui a produit la commutation, mais également selon que la commutation du signal de sortie est montante ou descendante. Ces retards seront notés  $T_{pLH}$  et  $T_{pHL}$  pour « Temps de Propagation Low High » et « Temps de Propagation High Low ».

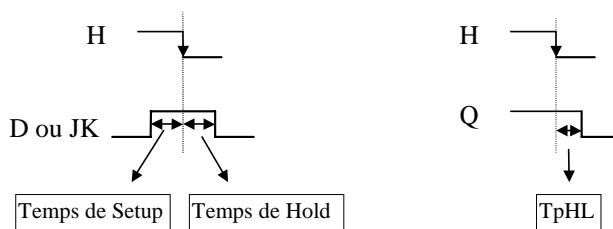


Figure 8.20. Paramètres temporels des bascules

## 8.2. Registres

Les registres sont des associations de bascules permettant de mémoriser et de réaliser certaines opérations sur des mots logiques.

### 8.2.1. Registre de mémorisation

Le registre de mémorisation est le registre élémentaire. Il est constitué d'une juxtaposition de bascules permettant de mémoriser un mot binaire (Figure 8.21). Ce registre est également appelé registre à entrées parallèles.

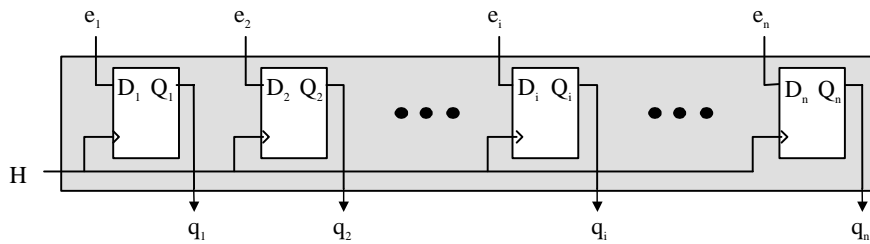


Figure 8.21. Register de mémorisation

### 8.2.2. Register de mémorisation avec signal d'inhibition

Le registre de mémorisation présenté précédemment n'est réellement utilisable dans la pratique que s'il est muni d'une entrée de contrôle permettant d'inhiber son chargement lorsque celui-ci n'est pas souhaité (cf §.8.4.). Le registre présenté sur la Figure 8.22 est un registre de mémorisation commandé par un signal d'inhibition C.

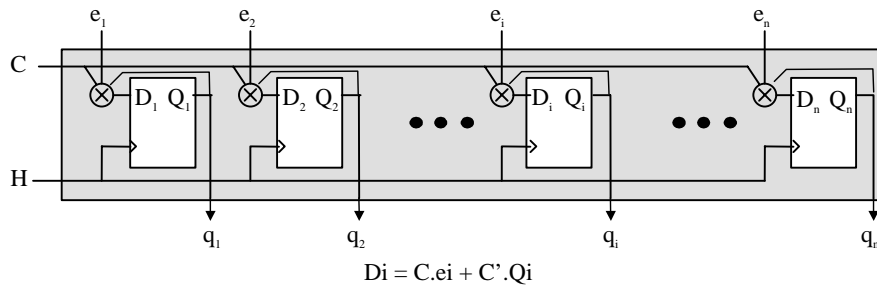


Figure 8.22. Register de mémorisation avec signal d'inhibition

### 8.2.3. Register à décalage

Le registre à décalage est une association de bascules permettant de décaler un mot binaire. L'entrée d'un mot dans le registre peut se faire, en fonction d'une commande C, soit par chargement parallèle comme précédemment soit par décalage à partir d'une entrée série. Le registre présenté sur la Figure 8.23 est un registre à décalage à droite. L'entrée  $e_1$  joue à la fois le rôle d'entrée parallèle et d'entrée série.

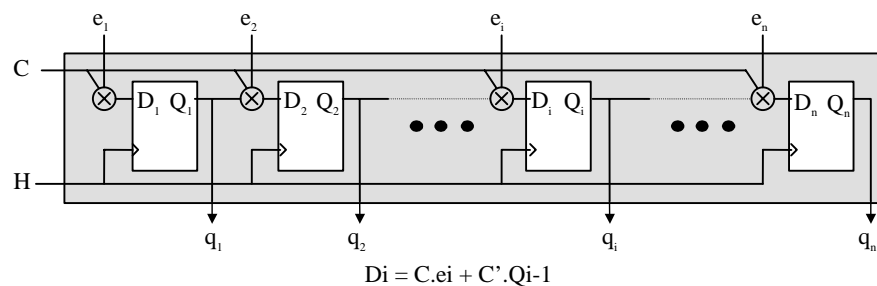


Figure 8.23. Register à décalage à droite

Remarque : Un registre à décalage à droite peut être utilisé comme un diviseur par 2 alors qu'un registre à décalage à gauche peut être utilisé comme un multiplieur par 2.

### 8.2.4. Registre universel

Le registre universel est une association de bascules permettant quatre modes de fonctionnement commandés par deux variables C1 et C2.

C1C2 = 00	Chargement parallèle
C1C2 = 01	Décalage à droite
C1C2 = 10	Décalage à gauche
C1C2 = 11	Inhibition de l'horloge.

Pour permettre ces quatre modes de fonctionnement, chacune des bascules est précédée d'un multiplexeur. L'entrée D de chaque bascule est ainsi fonction du mode de fonctionnement désiré (Figure 8.24).

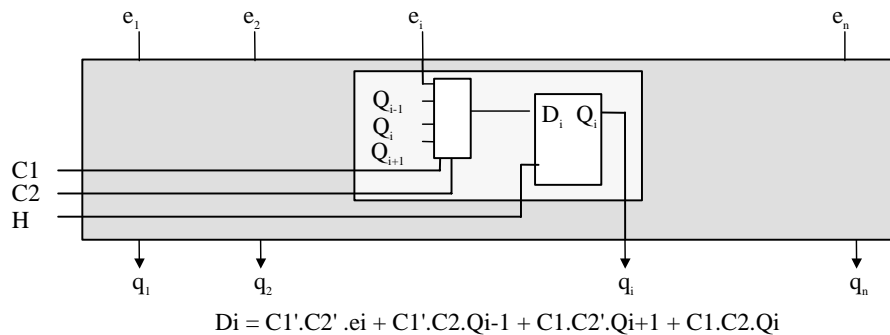


Figure 8.24. Registre universel

Remarque : L'entrée  $e_1$  peut jouer simultanément le rôle d'entrée parallèle et d'entrée série gauche. L'entrée  $e_n$  peut jouer simultanément le rôle d'entrée parallèle et d'entrée série droite.

## 8.3. Mémoires

Dans un ordinateur, les informations (programmes, données provenant de l'extérieur, résultats intermédiaires, données à transférer à l'extérieur, ...) doivent en général être conservées pendant un certain temps pour permettre leur exploitation. La conservation de ces informations est dévolue soit à des registres, soit à des structures plus adaptées aux grandes capacités de stockage : les mémoires vives (RAM).

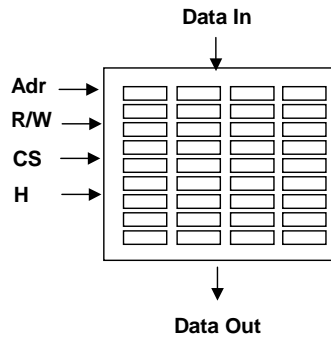
### 8.3.1. Mémoires vives

Les mémoires vives ou RAM sont des mémoires à lecture et écriture qui permettent d'enregistrer des informations, de les conserver et de les restituer. RAM signifie Random Access Memory. Littéralement cela se traduit par mémoire à accès aléatoire. En fait, l'accès à une RAM n'a rien d'aléatoire. Ce qu'il faut comprendre, c'est qu'on peut accéder à n'importe quelle partie de la mémoire directement, sans obligation technique particulière.

Dans une mémoire vive, l'information élémentaire, ou bit, est mémorisée dans une cellule ou point mémoire. Ces cellules sont groupées en mots de  $n$  bits, c'est-à-dire que les  $n$  bits sont traités (écrits ou lus) simultanément. Les cellules sont arrangées en bloc mémoire. L'organisation matricielle des blocs mémoires permet d'optimiser la structure tant d'un point de vue surface (adressage des mots) que temps d'accès. (éviter des pistes trop longues pour la distribution des différents signaux aux cellules).

Extérieurement, et en ne tenant compte que des signaux logiques, un bloc mémoire peut être représenté comme sur la figure 8.25. Pour pouvoir identifier individuellement chaque mot on utilise  $k$  lignes d'adresse

(signal adr). La taille d'un bloc mémoire est donc  $2^k$ , le premier mot se situant à l'adresse 0 et le dernier à l'adresse  $2^k - 1$ . Une commande (R/W) indique si la mémoire est accédée en écriture (l'information doit être mémorisée) ou en lecture (l'information doit être restituée). Une commande (CS) permet d'activer le fonctionnement de la mémoire (en lecture ou en écriture) ou de l'inhiber. Sur ce schéma on distingue deux canaux de n lignes en entrée et en sortie, mais dans d'autres cas les accès en entrée et en sortie peuvent être confondus en un seul canal bidirectionnel.



**Figure 8.25.** Architecture générale d'une mémoire

Parmi les caractéristiques d'une mémoire nous trouvons la capacité et le format. La capacité représente le nombre total de bits et le format correspond à la longueur des mots. Le nombre de bits d'adresse  $k$  définit le nombre total de mots de la mémoire, si  $n$  est le nombre de bits par mot, la capacité de la mémoire est donnée par :

$$\text{Capacité} = 2^k \text{ mots} = 2^k \times n \text{ bits}$$

Cette capacité est exprimée en multiple de 1024 ou kilo. La table suivante résume la valeur des autres préfixes utilisés pour exprimer les capacités des mémoires :

Symbole	Préfixe	Capacité
1 k	(kilo)	$2^{10} = 1024$
1 M	(méga)	$2^{20} = 1048576$
1 G	(giga)	$2^{30} = 1073741824$
1 T	(tera)	$2^{40} = 1099511627776$

La figure 8.26, présente une organisation logique possible pour une mémoire de 16 mots de  $n$  bits. Ici chaque mot est stocké dans une case de  $n$  bits, tel un registre. Cette case reçoit en entrée  $n$  lignes de données et une ligne de chargement. Elle dispose de  $n$  lignes de sortie fournissant le contenu du registre. Chacune de ces lignes est commandée par une porte "3 états". Ces cases sont organisées en une matrice de 4 lignes et 4 colonnes. Les 4 bits d'adresse (Adr) sont séparés en deux groupes, 2 bits pour identifier la ligne ( $A_L$ ) et 2 bits pour la colonne ( $A_C$ ). Les décodeurs de ligne et de colonne permettent de sélectionner les connexions à activer pour adresser la cellule souhaitée.

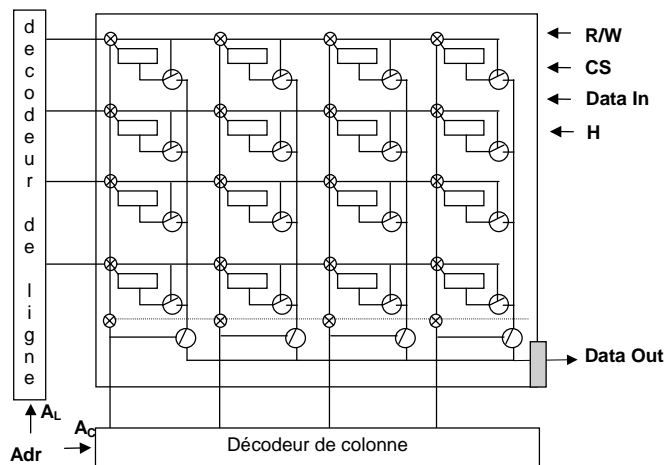


Figure 8.26. Architecture interne d'une mémoire

En lecture, les quatre cases de la ligne sélectionnée fournissent leur contenu sur les quatre bus verticaux. Une seule des quatre portes "3 états", au bas du schéma, est connectée à la sortie S du boîtier. Cette porte "3 états" fournit une amplification des signaux (registre de sortie).

En écriture, le mot à charger doit être présent sur l'entrée « Data In » du circuit. Ces données sont distribuées simultanément sur toutes les cellules de n bits. La ligne désignée par l'adresse  $A_L$  est à 1. Le signal de chargement est transmis à la seule colonne identifiée par l'adresse  $A_C$ . Seul le registre à l'intersection de cette ligne et de cette colonne est donc chargé.

### 8.3.2. Mémoires RAM Statiques / Dynamiques

Il existe deux grandes familles de mémoires RAM : les RAM statiques (SRAM) ou les RAM dynamiques (DRAM).

Dans le cas des RAM statique, le point mémoire élémentaire est une bascule (Figure 7.27). Les bascules garantissent la mémorisation de l'information aussi longtemps que l'alimentation électrique est maintenue sur la mémoire.

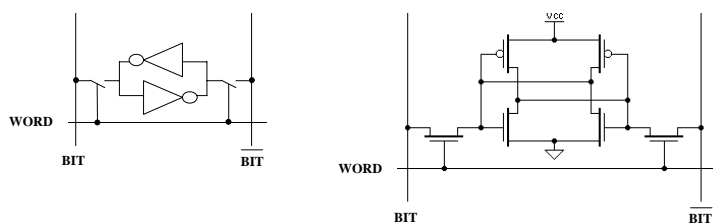


Figure 8.27. Cellule mémoire SRAM

Dans le cas des mémoires dynamiques (DRAM), l'élément de mémorisation est un condensateur (capacité) commandée par un transistor (Figure 8.28). L'information est mémorisée sous la forme d'une charge électrique stockée dans le condensateur. Cette technique permet une plus grande densité d'intégration, car un point mémoire nécessite environ deux à quatre fois moins de place que dans une mémoire statique. Par contre, du fait des courants de fuite le condensateur a tendance à se décharger. C'est pourquoi ces mémoires doivent être rafraîchies régulièrement pour entretenir la mémorisation : il s'agit de lire l'information avant qu'elle n'ait

totallement disparu et de la recharger. Ces mémoires sont dites RAM dynamique (DRAM) du fait de cette opération de rafraîchissement.

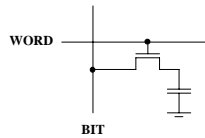


Figure 8.28. Cellule mémoire DRAM

Ce rafraîchissement indispensable a plusieurs conséquences. Tout d'abord il complique la gestion des mémoires dynamiques car il faut tenir compte des actions de rafraîchissement qui sont prioritaires. D'autre part, la durée de ces actions augmente le temps d'accès aux informations. Le temps d'attente des données est variable selon que la lecture est interrompue ou non par des opérations de rafraîchissement et la quantité de cellules à restaurer. Il faut donc se placer dans le cas le plus défavorable pour déterminer le temps d'accès à utiliser en pratique.

En général les mémoires dynamiques, qui offrent une plus grande densité d'information et un coût par bit plus faible, sont utilisées pour la mémoire centrale, alors que les mémoires statiques, plus rapides, sont utilisées pour les caches.

## 8.4. Compteurs / décompteurs

### 8.4.1. Définitions

Un compteur est une association de  $n$  bascules permettant de décrire, au rythme d'une horloge, une séquence déterminée qui peut avoir au maximum  $2^n$  combinaisons différentes. Les combinaisons apparaissent toujours dans le même ordre.

**Définition** : Une combinaison de sortie d'un compteur est appelée *état*. Le nombre d'états différents pour un compteur est appelé le *modulo*  $\sim$  de ce compteur.

Un compteur modulo  $N$  démarrant à 0 et comptant dans l'ordre binaire naturel compte de 0 à  $N-1$ . Le graphe présenté sur la figure 29 est le graphe d'un compteur binaire modulo 8.

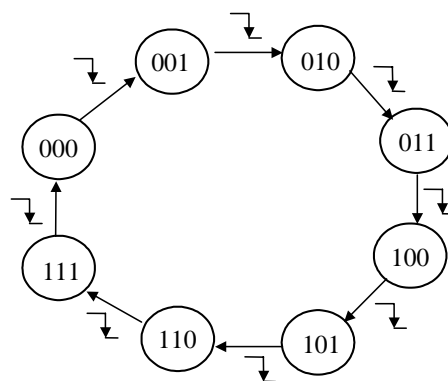


Figure 8.29. Graphe d'un compteur modulo 8

8.4.2. Compteurs « asynchrones »

Ce type de structure est présenté ici pour des raisons pédagogiques, mais comme nous le verrons au cours de cet exposé il n'est absolument pas à recommander car pouvant être la source de nombreux ennuis.

Pour construire un compteur, nous pouvons remarquer qu'une bascule T dont l'entrée est à 1 fonctionne en diviseur de fréquence. Il en est de même que lorsque la sortie complémentée d'une bascule D est rebouclée sur l'entrée D ou que les entrées d'une bascule JK sont égales à 1, ces bascules (Figure 8.30).

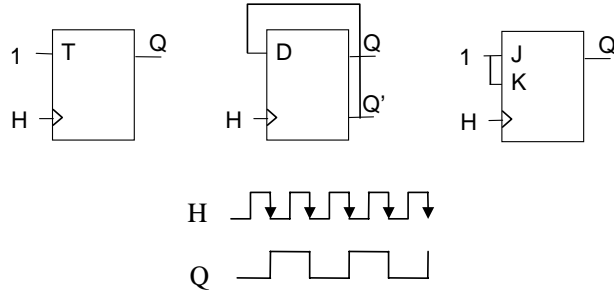


Figure 8.30. Bascules montées en diviseur de fréquence

8.4.2.a. Compteurs « asynchrones » modulo  $2^n$

En cascade les bascules selon le schéma de la Figure 8.31, on réalise un dispositif répondant au cahier des charges fonctionnel d'un compteur modulo  $2^n$  ( $n$  étant le nombre de bascules). Ce compteur est appelé compteur « asynchrone » du fait que toutes les bascules ne sont pas commandées par le même signal d'horloge. Le chronogramme correspondant est présenté sur la Figure 8.32.

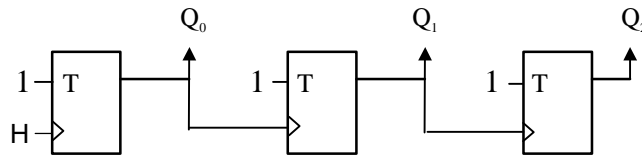


Figure 8.31. Compteur modulo 8 asynchrone

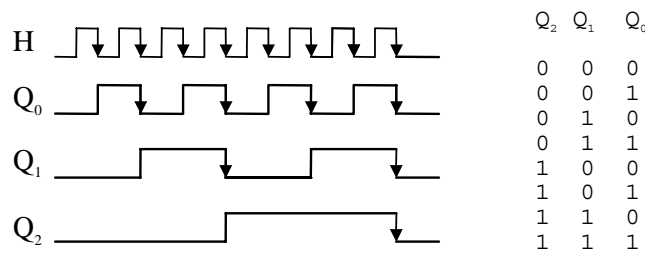


Figure 8.32. Chronogramme du compteur par 8

Pour réaliser un décompteur il suffit de considérer sur les sorties  $Q'$  des bascules (Figure 8.33) ou de réaliser le même montage avec des bascules fonctionnant sur front montant (Figure 8.34).

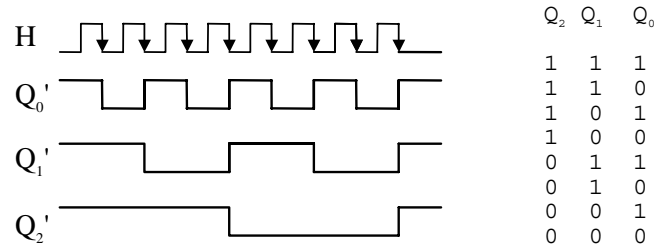


Figure 8.33. Chronogramme du décompteur par 8

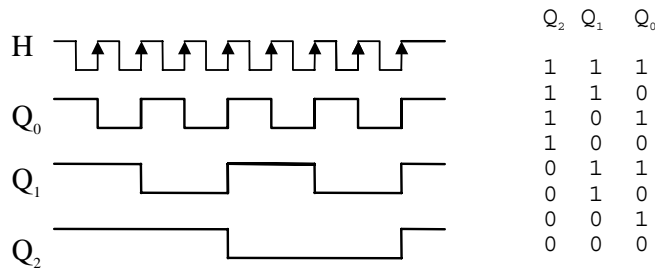


Figure 8.34. Décompteur par 8 avec bascules fonctionnant sur fronts montants

#### 8.4.2.b. Compteurs « asynchrones » modulo différent de $2^n$

Pour réaliser un compteur ou un décompteur dont le modulo n'est pas une puissance de 2, une solution qui pourrait être envisagée est d'agir sur l'entrée « Clear » lorsque la combinaison correspondant au modulo du compteur ce produit sur les sorties de celui ci.

Exemple : La Figure 8.35 présente la structure d'un compteur asynchrone par 6 ainsi que le chronogramme associé. Pour réaliser un compteur par 6, il suffit de détecter la combinaison  $Q_2Q_1Q_0 = 110$  et de la renvoyer sur le signal « Clear »

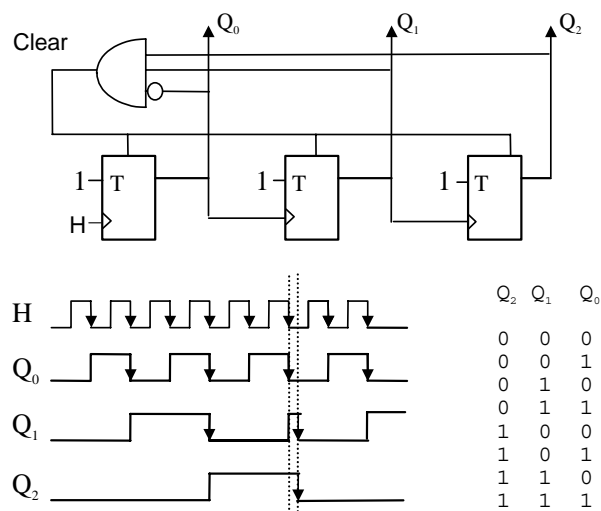


Figure 8.35. Compteur asynchrone par 6

Si la démarche suivie pour construire ces compteurs (décompteurs) peut paraître attrayante, les structures obtenues présentent des inconvénients majeurs qui vont jusqu'à remettre en cause le fonctionnement des structures précédemment présentées.

Les bascules ne commutant pas sur le même signal d'horloge, les retards de commutation se cumulent sur chacune des bascules du compteur. En effet, c'est la commutation de la première bascule qui entraîne l'activation de la seconde qui elle-même entraîne l'activation de la troisième, etc. Ainsi, la fréquence maximum de fonctionnement  $F_H$  d'un compteur modulo  $n$ , constitué de  $n$  bascules de délai de propagation  $D_p$  dépend du nombre de bascules du compteur et donc du modulo du compteur. Cette fréquence peut être établie comme suit :

$T_{max} = D_p * n$	Délai de propagation du compteur
$T_H \geq T_{max}$	Période de l'horloge
$F_H \leq 1/(T_{max}) = 1/(n * D_p)$	Fréquence de l'horloge

D'autre part, ces retards de commutation introduisent des états transitoires relativement conséquents (Figure 8.36).

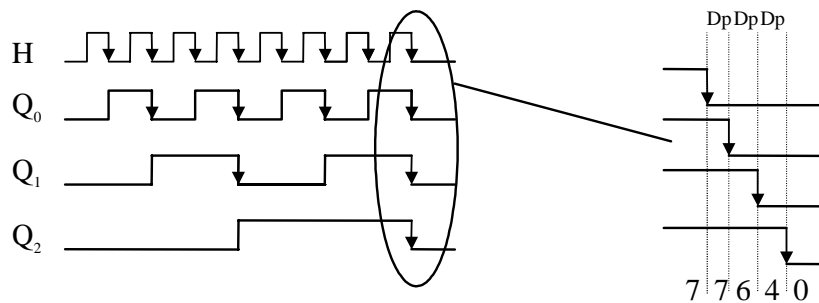


Figure 8.36. Etats transitoires

Mais l'inconvénient majeur est lié au fait que cette structure nécessite de la logique sur des signaux asynchrones (Horloge générée par une bascule et Clear généré par une structure combinatoire). Cette logique peut engendrer (ou propager) des transitoires ou des courses critiques entre signaux d'Horloge et de Clear qui peuvent entraîner des dysfonctionnements du compteur. Ce type de structures, qui a été présenté pour raisons pédagogiques, est donc à proscrire impérativement.

### 8.4.3. Compteurs synchrones

Un compteur synchrone est une structure où toutes les bascules reçoivent le même signal d'horloge (Figure 8.37). La fonction comptage ou décomptage est réalisée par l'intermédiaire des fonctions appliquées sur les entrées synchrones des bascules.

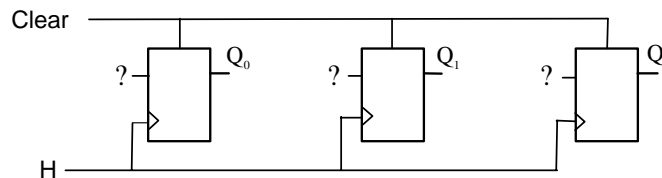


Figure 8.37. Structure générale d'un compteur par 8 synchrone réalisé avec des bascules JK

8.4.3.a. Compteurs « synchrones » modulo  $2^n$ 

Pour que le compteur décrive une séquence déterminée, il faut commander les entrées des bascules (T, D ou JK) de façon adéquate. Pour cela, on peut remarquer sur la table de vérité du compteur (Figure 8.38) que le bit de poids faible change à tous les coups d'horloge et que qu'un bit quelconque change lorsque tous les bits de droite sont égaux à 1.

$Q_2$	$Q_1$	$Q_0$
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Figure 8.38. Table de vérité d'un compteur par 8

Sachant que sur une bascule T (JK), il y a inversion de la sortie pour  $T = 1$  ( $JK = 11$ ), on peut en déduire les entrées de chacune des bascules et par conséquent la structure des compteurs synchrones (Figure 8.39).

$$T_0 = 1$$

$$T_1 = Q_0$$

$$T_2 = Q_0 \cdot Q_1$$

$$T_n = Q_0 \cdot Q_1 \dots Q_{n-1}$$

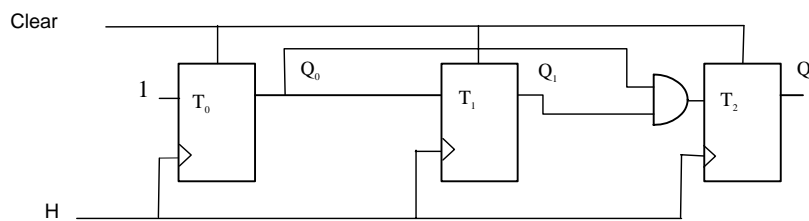


Figure 8.39. Compteur par 8 synchrone

Le raisonnement fait précédemment avec des bascules T (JK) peut être mené à l'identique avec des bascules D sachant que les deux structures présentées sur la figure 8.40 sont fonctionnellement parfaitement équivalentes.

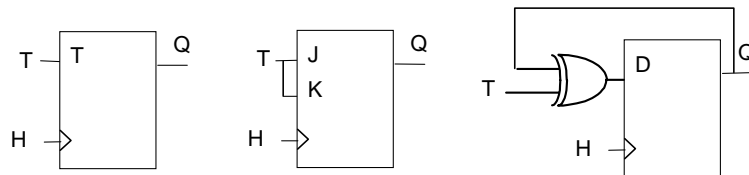


Figure 8.40. Bascules T, JK, D

La fréquence maximum de fonctionnement  $F_H$  d'un compteur modulo  $n$ , constitué de  $n$  bascules de délai de propagation  $D_p$  peut être établie comme suit :

$$T_{\max} = D_p$$

Délai de propagation du compteur

$$T_H \geq T_{\max}$$

Période de l'horloge

$$F_H \leq 1/(T_{max}) = 1/Dp$$

Fréquence de l'horloge

Le compteur synchrone est donc plus rapide que le compteur asynchrone puisque les délais de propagation des bascules ne sont pas cumulés. D'autre part, si l'on suppose que toutes les bascules ont le même délai de propagation il n'y a pas d'état transitoire sur la sortie. Dans la pratique, ce n'est bien évidemment pas le cas car les délais de propagation de bascules peuvent être différents (temps de montée et de descente différents, charges différentes etc.). Toutefois, la durée de ces transitoires est réduite à la différence de fonctionnement des bascules et en aucun cas n'est aussi importante qu'en asynchrone.

De la même manière que dans le cas asynchrone, un décompteur peut être obtenu en sortant sur les sorties  $Q'$  du compteur. On peut également réaliser un décompteur en remarquant sur la table de vérité (Figure 8.41) que le bit de poids faible change à tous les coups d'horloge et que qu'un bit quelconque change lorsque tous les bits de droite sont égaux à 0.

$Q_2$	$Q_1$	$Q_0$
1	1	1
1	1	0
1	0	1
1	0	0
0	1	1
0	1	0
0	0	1
0	0	0

Figure 8.41. Table de vérité d'un décompteur par 8

Sachant que sur une bascule T, il y a inversion de la sortie pour  $T=1$ , on peut en déduire les entrées de chacune des bascules et par conséquent la structure des décompteurs synchrones (Figure 8.42).

$$T_0 = 1$$

$$T_1 = Q_0'$$

$$T_2 = Q_0' \cdot Q_1'$$

$$T_n = Q_0' \cdot Q_1' \cdot \dots \cdot Q_{n-1}'$$

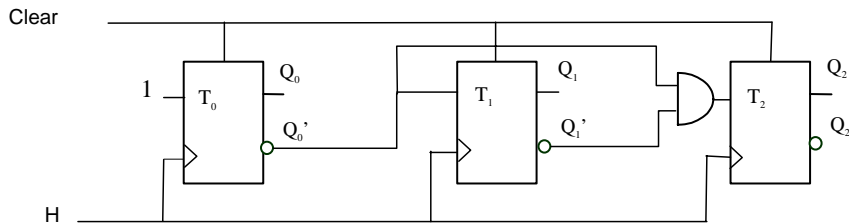


Figure 8.42. Décompteur par 8 synchrone

Par le même raisonnement, on peut déterminer la structure d'un compteur / décompteur synchrone (Figure 8.43) dont le mode comptage ou décomptage est commandé par une commande C ( $C=0 \Rightarrow$  Comptage,  $C=1 \Rightarrow$  Décomptage).

$$T_0 = 1$$

$$T_1 = C' \cdot Q_0 + C \cdot Q_0' = C \oplus Q_0$$

$$T_2 = C' \cdot Q_0 \cdot Q_1 + C \cdot Q_0' \cdot Q_1'$$

$$T_n = C' \cdot Q_0 \cdot Q_1 \cdot \dots \cdot Q_{n-1} + C \cdot Q_0' \cdot Q_1' \cdot \dots \cdot Q_{n-1}'$$

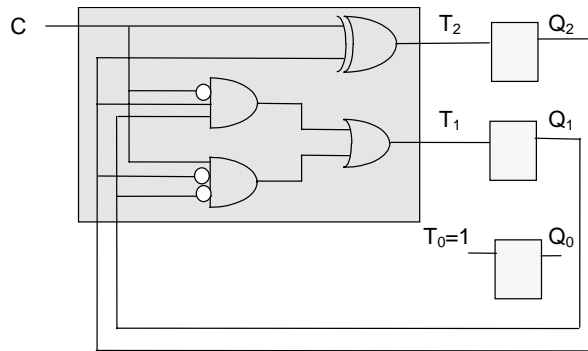


Figure 8.43. Schéma logique du compteur / décompteur par 8

**Remarque** : Le système précédent permet de basculer du mode comptage au mode décomptage et inversement sans modification de l'état courant. Ce ne peut être le cas d'un système basé sur fonctionnement asynchrone ou d'un système basé sur un compteur synchrone avec sortie Q et Q' multiplexées.

#### 8.4.3.b. Compteurs « synchrones » modulo différent de $2^n$

Pour réaliser un compteur, un décompteur ou un compteur / décompteur dont le cycle n'est pas une puissance de 2, on pourrait envisager, comme en asynchrone, d'agir sur l'entrée « Clear » lorsque la combinaison correspondant au modulo du compteur ce produit sur les sorties de celui ci. Mais, comme il a été précisé précédemment, cette solution est à proscrire absolument car les transitoires produits ou transmis par la logique sur le signal asynchrone de Clear risquent d'entraîner un dysfonctionnement de la structure.

Une autre solution permettant de réaliser un compteur, un décompteur ou un compteur / décompteur dont le cycle n'est pas une puissance de 2, est de redéfinir les fonctions d'entrée des bascules pour réaliser la fonction souhaitée.

##### Exemple 1 : Compteur par 6.

Le fonctionnement du compteur par 6 doit rester identique à celui du compteur par 8 tant que la combinaison 5 n'est pas présente sur les sorties du compteur. Lorsque la combinaison 5 est présente, les fonctions d'entrée des bascules doivent être modifiées. Au lieu de passer de 101 à 110, il faut passer de 101 à 000.

Soit  $C_5$  un flag prévenant qu'on est sur la combinaison 5.  $C_5 = Q_2 \cdot Q_1 \cdot Q_0$

En reconsidérant le fonctionnement des bascules lorsque  $C_5$  vaut 1 on obtient :

$T_0 = 1$  (Même fonctionnement que  $C_5$  vaille 0 ou 1)

$T_1 = C_5' \cdot Q_0 + C_5 \cdot 0 = C_5' \cdot Q_0$  (Conservation de  $Q_1$  lorsque  $C_5=1$ )

$T_2 = C_5' \cdot Q_0 \cdot Q_1 + C_5 \cdot 1 = C_5' \cdot Q_0 \cdot Q_1 + C_5$  (Inversion de  $Q_2$  lorsque  $C_5=1$ )

##### Exemple 2 : Compteur / décompteur par 6.

Pour le mode comptage ( $C=0$ ), le fonctionnement doit rester identique au compteur par 8 tant que la combinaison 5 n'est pas présente sur les sorties du compteur. Lorsque la combinaison 5 est présente, les fonctions d'entrée des bascules doivent être modifiées. Au lieu de passer de 101 à 110, il faut passer de 101 à 000.

Pour le mode décomptage ( $C=1$ ), le fonctionnement doit rester identique au décompteur par 8 tant que la combinaison 0 n'est pas présente sur les sorties du compteur. Lorsque la combinaison 0 est présente, les fonctions d'entrée des bascules doivent être modifiées. Au lieu de passer de 000 à 111, il faut passer de 000 à 101.

Soit  $C_0$  un flag prévenant qu'on est sur la combinaison 0.  $C_0 = Q_2' \cdot Q_1' \cdot Q_0'$

Soit  $C_5$  un flag prévenant qu'on est sur la combinaison 5.  $C_5 = Q_2 \cdot Q_1' \cdot Q_0$

Les équations des entrées de bascules peuvent s'exprimer de la manière suivante :

$$T_0 = C'[1] + C[1] = 1$$

$$T_1 = C'[C_5' \cdot Q_0] + C[C_0' \cdot Q_0']$$

$$T_2 = C'[C_5' \cdot Q_0 \cdot Q_1 + C_5] + C[C_0' \cdot Q_0' \cdot Q_1' + C_0]$$

#### 8.4.3.c. Compteurs « synchrones » avec signal d'inhibition

La prise en compte d'un signal d'inhibition (Inib) du compteur peut se faire de la même manière, c'est à dire en intervenant sur les entrées synchrones ( $T_i$ ) des bascules.

Exemple : Compteur / décompteur par 6 avec signal d'inhibition (Inib)

Les équations des entrées des bascules T peuvent s'exprimer de la manière suivante :

$$T_0 = \text{Inib}'$$

$$T_1 = \text{Inib}' \{ C'[C_5' \cdot Q_0] + C[C_0' \cdot Q_0'] \}$$

$$T_2 = \text{Inib}' \{ C'[C_5' \cdot Q_0 \cdot Q_1 + C_5] + C[C_0' \cdot Q_0' \cdot Q_1' + C_0] \}$$

#### 8.4.3.d. Comparaison bascules T / bascules D

Les bascules T sont particulièrement intéressantes pour la réalisation de compteurs car contrairement aux bascules D, elle ne nécessitent pas d'exprimer les condition de maintien de des sorties de bascules. En effet, seules sont exprimées les conditions de commutation, ce qui simplifie de manière notoire les fonctions d'entrée de bascule.

A titre de comparaison, le compteur/décompteur par 6 avec signal d'inhibition (Inib) précédent conduirait, avec des bascules D, aux équations suivantes :

$$D_0 = \text{Inib}' \{ C[C_5' \cdot Q_0' + C_5 \cdot Q_0'] + C'[C_0' \cdot Q_0' + C_0 \cdot Q_0'] \} + \text{Inib} \cdot Q_0$$

$$= \text{Inib}' \cdot Q_0' + \text{Inib} \cdot Q_0$$

$$= \text{Inib} \oplus Q_0'$$

$$D_1 = \text{Inib}' \{ C[C_5' \cdot (Q_0 \cdot Q_1' + Q_0' \cdot Q_1) + C_5 \cdot 0] + C'[C_0' \cdot (Q_0' \cdot Q_1' + Q_0 \cdot Q_1) + C_0 \cdot 0] \} + \text{Inib} \cdot Q_0$$

$$= \text{Inib}' \{ C[C_5' \cdot (Q_0 \oplus Q_1)] + C'[C_0' \cdot (Q_0 \oplus Q_1')] \} + \text{Inib} \cdot Q_1$$

$$D_2 = \text{Inib}' \{ C[C_5' \cdot (Q_0 \cdot Q_1 \oplus Q_2)] + C'[C_0' \cdot (Q_0 + Q_1 \oplus Q_2')] \} + \text{Inib} \cdot Q_2$$

Bien qu'il soit probable que, dans ce cas particulier, les équations puissent encore se simplifier elles restent de toute manière plus complexes que celles obtenues avec des bascules T.

Remarque : Ces équations sont obtenues par un raisonnement généralisable à des compteurs de modulo différent.

## 8.5. Règles de conception

Une plaie de trop de réalisations rencontrées est le mélange, dans une même unité fonctionnelle, des commandes asynchrones et synchrones. En effet, la présence de commandes asynchrones dans une conception synchrone induit une sensibilité aux phénomènes transitoires et/ou parasites qui peut être source de nombreux ennuis. D'autre part, la conception d'une application complexe n'est généralement envisageable qu'au travers d'un processus initial de partitionnement du cahier des charges.

### 8.5.1. Signaux de forçage asynchrones

Les signaux de forçage à action directe (Clear ou Preset), c'est à dire indépendants de l'horloge (asynchrone) peuvent servir à initialiser une application par une commande spécifique mais toute autre utilisation notamment pour réaliser une fonction particulière (commande par de la logique) peut être la source de nombreux ennuis. En effet, toute logique sur ces signaux peut engendrer des impulsions transitoires de durées inconnues, souvent très faibles, mais suffisante pour conduire à un résultat qui, s'il peut être instructif dans un contexte d'enseignement, est catastrophique dans une réalisation. Dans ce cas, une carte qui semble donner toute satisfaction quand on l'observe avec un oscilloscope, peut par exemple cesser de fonctionner dès que l'on retire l'appareil de mesure. Ce phénomène provient généralement de la modification de la durée ou de l'amplitude d'impulsions transitoires sur les signaux asynchrones induite par la charge capacitive supplémentaire apportée par la sonde de mesure. Le dépannage d'un tel système relève alors plus de la divination que d'une méthodologie raisonnée. Ce type de pratique est donc à condamner sans appel.

### 8.5.2. Les signaux d'horloges

Le blocage, par exemple par une porte, des signaux d'horloge pour maintenir l'état d'un registre, est une autre erreur que l'on rencontre parfois. Cette faute, qui provoque des décalages temporels entre les signaux d'horloge (*clock skew*) appliqués aux différentes parties d'une carte, ou d'un circuit, risque de conduire à des violations de temps de maintien ou de pré-positionnement, d'où des comportements imprévisibles des registres concernés.

Un autre effet pervers des circuits combinatoires de « calcul » des signaux d'horloge, est la génération, difficile à contrôler, d'impulsions parasites sur ces signaux. La recherche de ces impulsions, suffisamment larges pour faire commuter les circuits actifs sur des fronts, mais suffisamment étroites pour ne pas être vues lors d'un examen rapide avec un oscilloscope, est un passe temps dont on se lasse très vite.

Quand il est nécessaire d'appliquer à différentes parties d'un ensemble des signaux d'horloges différents, il est indispensable de traiter à part, et de façon méticuleuse, la réalisation du distributeur d'horloge correspondant. Notons, en passant, que pour ces fonctions il convient de surveiller de très près les modifications apportées par les optimiseurs ; ces derniers ont la fâcheuse tendance d'éliminer les portes inutiles d'un point de vue algébrique, même si elles sont utiles d'un point de vue circuit.

### 8.5.3. Conception synchrone

S'il y a une règle de conception à retenir des deux paragraphes précédent pour éviter tout désagrément du aux phénomènes temporels et transitoires c'est d'éviter à tout prix d'intervenir sur les signaux asynchrones que sont l'horloge (H) et les signaux de forçage (Clear, Preset). Ainsi, les seuls signaux sur lesquels on doit intervenir pour réaliser la fonction souhaitée sont les entrées synchrones (D) des bascules. Cette règle de conception est illustrée sur la figure 8.44.

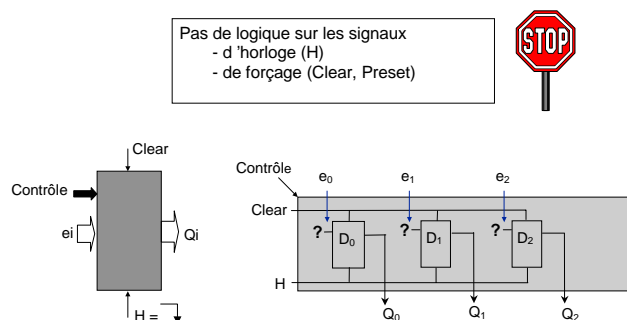


Figure 8.44. Conception synchrone

Tout manquement à cette règle peut conduire à de sérieux désagréments ou en tout cas faire l'objet d'une attention toute particulière.

#### **8.5.4. Diviser pour régner**

L'erreur de méthode la plus fréquente, et la plus pénalisante, que commettent beaucoup de débutants dans la conception des systèmes électroniques, qu'ils soient analogiques ou numériques, est sans doute de dessiner des schémas, voire de les câbler, avant même d'avoir une vision claire de l'ensemble de la tâche à accomplir.

Le travail de réflexion sur la structure générale d'une application est primordial. Ce que l'on appelle traditionnellement la méthode descendante (*top down design*), n'est rien d'autre que l'application de cette règle simple : quand on conçoit un ensemble, *on va du général au particulier*, on ne s'occupe des détails que quand le cahier des charges a été mûrement réfléchi, et que le plan général de la solution a été établi. Si, au cours de la descente vers les détails, on découvre qu'une difficulté imprévue apparaît, il faut revenir au niveau général pour voir comment la réponse à cette difficulté s'insère dans le plan d'ensemble.

Le premier réflexe à avoir, face à un problème, un tant soit peu complexe à résoudre, est de le couper en deux. La démarche précédente est répétée, pour chaque demi-problème, jusqu'à obtenir des sous-ensembles dont la réalisation tient en quelques circuits élémentaires, en quelques lignes de code source dans un langage ou dans un diagramme de transitions qui ne dépasse pas une dizaine d'états différents.