4.3 Notions de registres

Un registre est un ensemble de cellules mémoires constituées par des bascules.
Le contenu d'un registre peut donc être considéré comme un nombre binaire ou un "mot" de n bits.

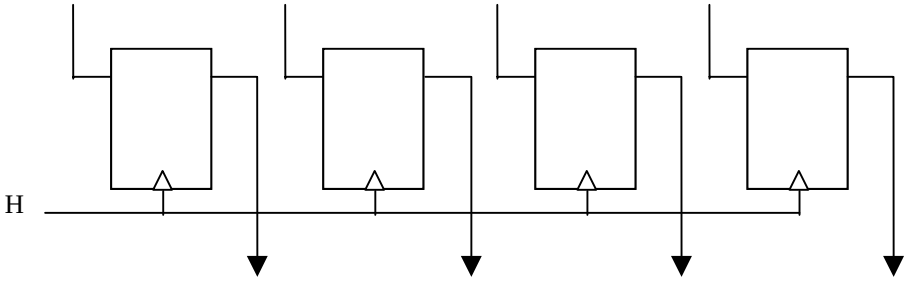
Exemple : 0 1 1 0, chacun des bits sera stocké par une bascule.

Les applications des registres sont nombreuses :

- conversion série – parallèle
- multiplication ou division par une puissance de 2
- ligne à retard numérique

4.3.1 Registre à mémoire

La fonction d'un tel registre est de "stocker / mémoriser" un mot de n bits.
Exemple de réalisation (à l'aide de bascules D edge) : chacune des cellules est une bascule D.

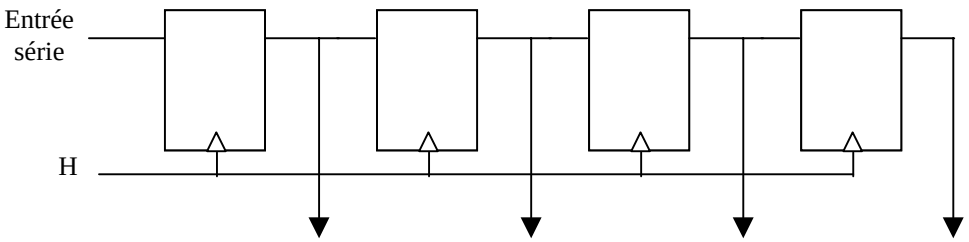


La sortie du registre mémorise le mot d'entrée tant que H = 0 ou 1. Lorsque l'horloge présente un front montant, les données en sortie sont actualisées. Le registre peut être initialisé grâce aux entrées de forçage asynchrone qui peuvent forcer les sorties des bascules à 0 ou à 1.

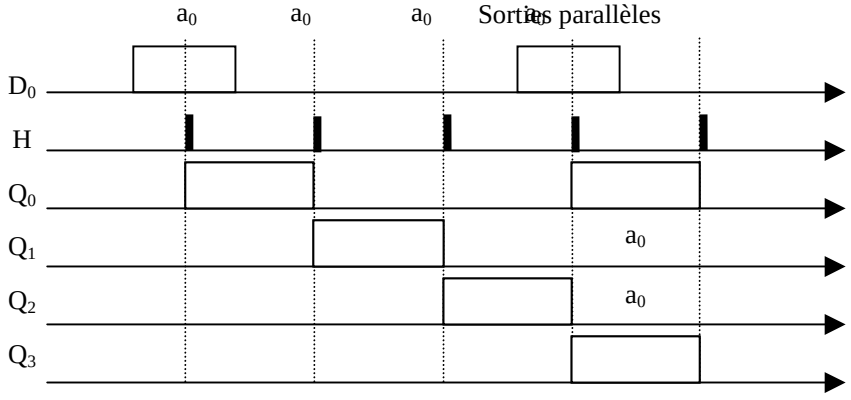
4.3.2 Registre à décalage – conversion série/parallèle

Ce type de registre sert à décaler tous les bits d'un mot de un ou plusieurs crans vers la droite ou vers la gauche. Ces systèmes peuvent être utilisés pour effectuer des multiplications ou divisions par une puissance de 2, ou encore pour effectuer une conversion série – parallèle.

Exemple de réalisation :



D'où le chronogramme :



Remarque :
Les bascules doivent être commandées sur front pour contrôler l'instant du décalage (un bit à la fois et non plusieurs si la durée du niveau est trop grande)

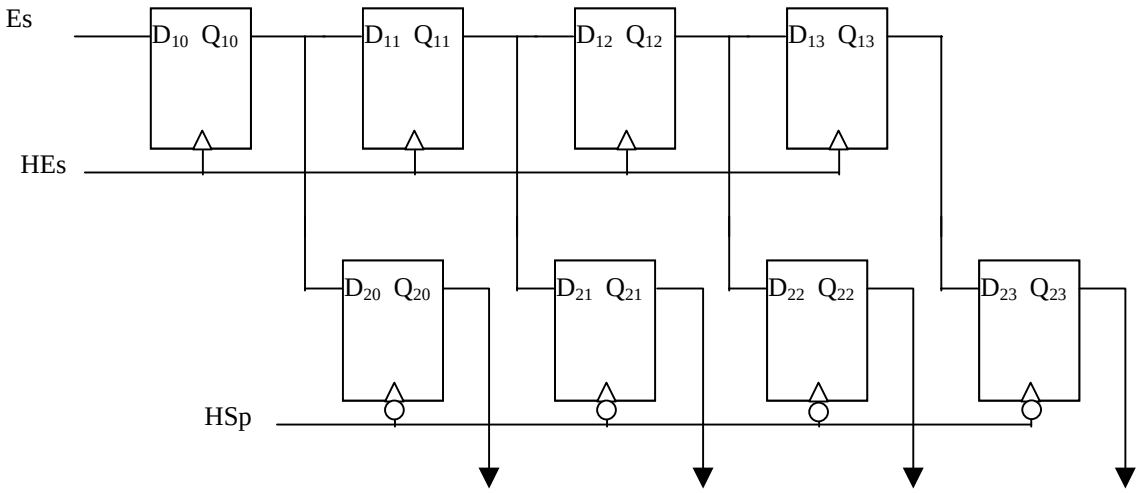
Exemple d'application : **la multiplication par 2^n**

Soit N = 3 = 0011
2.N = 6 = 0110
2.(2.N) = 12 =1100

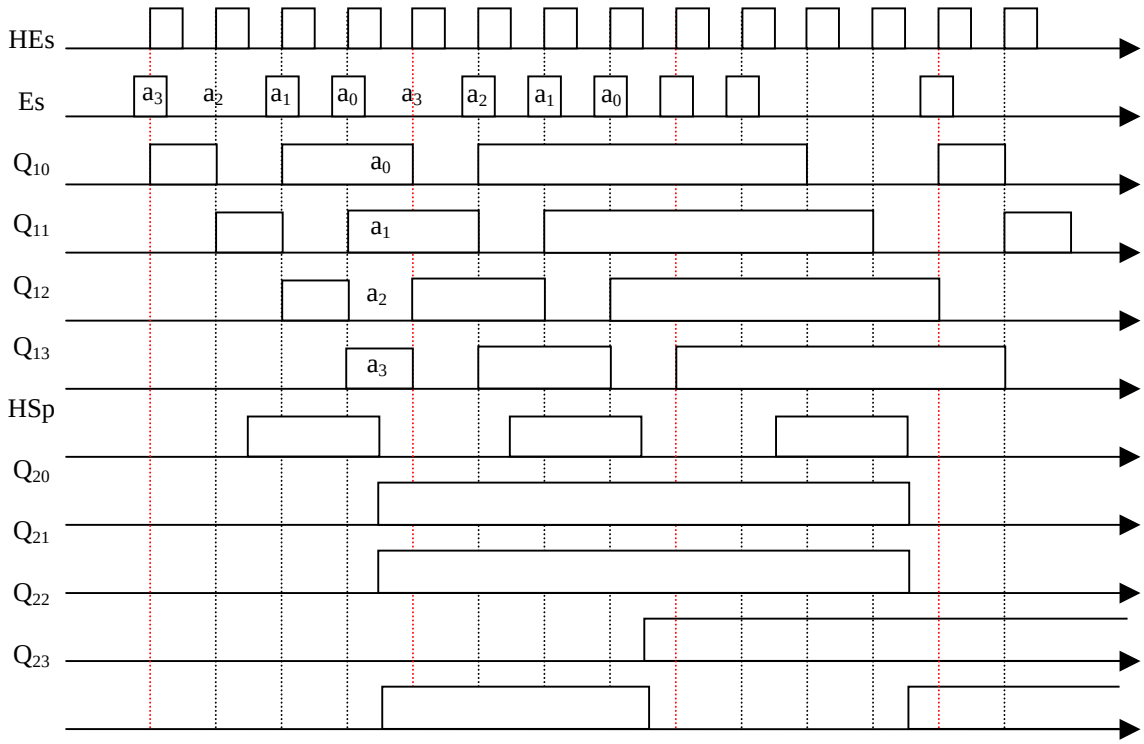
On constate que pour effectuer la multiplication d'un nombre par 2 il suffit de décaler tous les bits du nombre de 1 cran vers la droite (vers les bits de poids fort - MSB).
De la même façon, pour réaliser la division d'un nombre par 2 il suffit de décaler tous les bits du nombre de 1 cran vers la gauche (vers les bits de poids faible - LSB).

4.3.3 Exemple de conversion série – parallèle

Le but est ici de réaliser un système capable de convertir les 4 bits série d'un mot sous la forme de 4 bits disponibles simultanément. Par ailleurs on désire que les sorties de ce système ne puissent être "rafraîchies" qu'à des instants déterminés.
Il faudra donc utiliser un registre à décalage pour remplir la 1ère fonction et un registre à mémoire pour remplir la 2nde.
On supposera que les bits de poids fort arrivent avant les bits de poids faibles.



Chronogramme :



4.3 Compteurs

Un compteur est un élément que l'on utilise lorsque l'on a besoin d'établir une relation d'ordre sur une succession d'événements.
Bien sûr il sert également à effectuer la fonction de comptage d'événements, mais il a des applications très variées, comme par exemple la division de fréquence.
Un compteur comprend une entrée et plusieurs sorties. Les sorties présentent une combinaison de bits à 0 et à 1 qui correspond à un codage du nombre d'impulsions injectées en entrée.

Exemple : compteur par 8 binaire.

Nombre d'impulsions	Sorties (binaires)			comptage
0	0	0	0	0
1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	1	0	1	5
6	1	1	0	6
7	1	1	1	7
8	0	0	0	0
9	0	0	1	1
10	0	1	0	2

On obtient en sortie le nombre d'impulsions écoulées modulo 8. Pour cela on utilise le terme « *compteur modulo 8* ».

4.3.1 Compteur cyclique ou en anneau

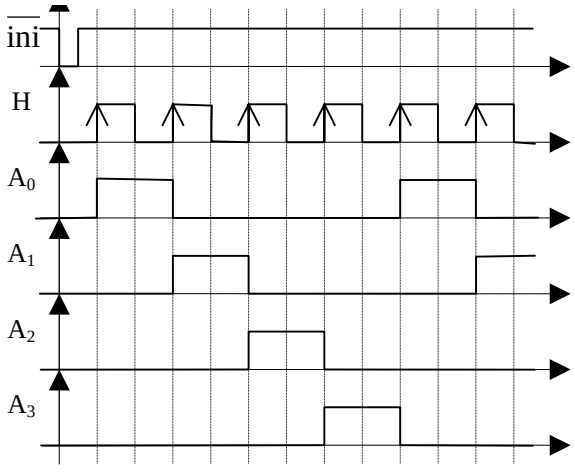
Dans ce type de compteur **le nombre de bascules est égal au modulo du compteur** (exemple : compteur modulo 8 => 8 bascules). L'information est constituée par la position d'un ou plusieurs bits à 1 dans les bascules.

Exemple : compteur en anneau modulo 4. Ce compteur est un registre à décalage avec une entrée parallèle pour l'initialisation.

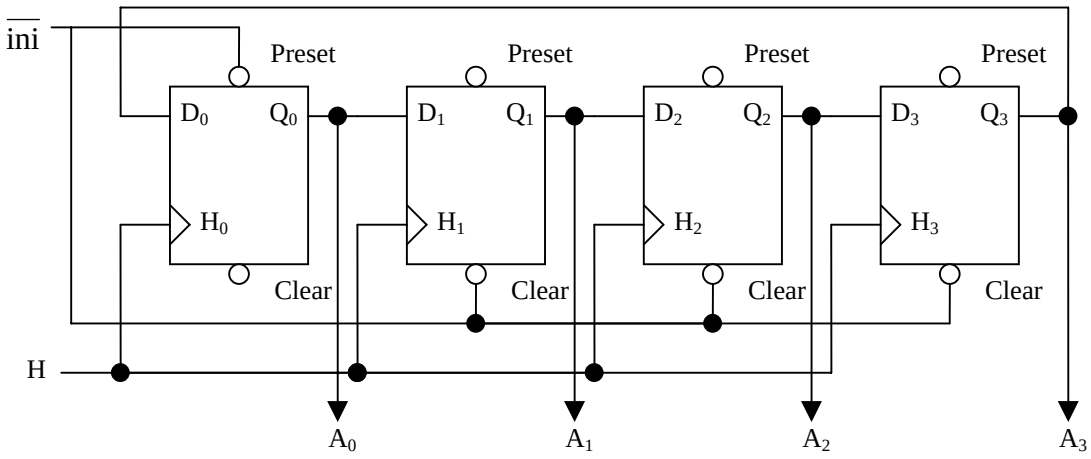
- *Table de fonctionnement :*

N	A_0	A_1	A_2	A_3
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0

- *Chronogramme :*



- *Schéma de principe :*



4.3.1 Compteur asynchrone modulo 2^n

La logique asynchrone est en général plus rapide que la logique synchrone, mais elle souffre de deux inconvénients majeurs :

- Une mise en œuvre plus délicate car moins systématique ;
- Une testabilité parfois difficile.

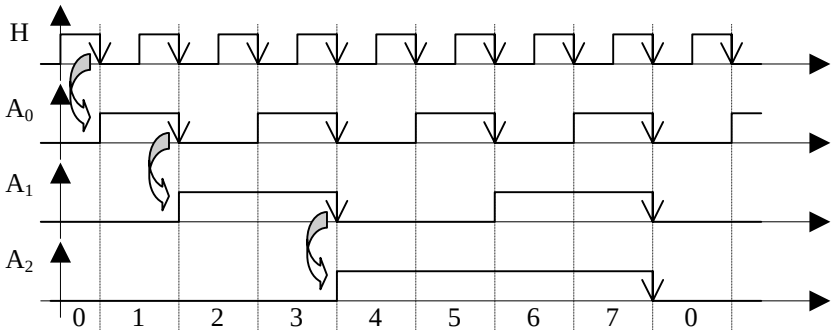
En pratique, les compteurs sont réalisés de plus en plus en logique synchrone, sauf lorsque le comptage se fait modulo 2^n.

Un compteur asynchrone modulo 2^n peut être réalisé soit à partir de bascules T, D ou JK (avec J=K=1). Nous donnons ci-dessous un exemple de réalisation à l'aide de bascules D pour un comptage modulo 2^3. Dans ce cas le nombre de bascules est égal à 3 (=n du modulo 2^n du compteur).

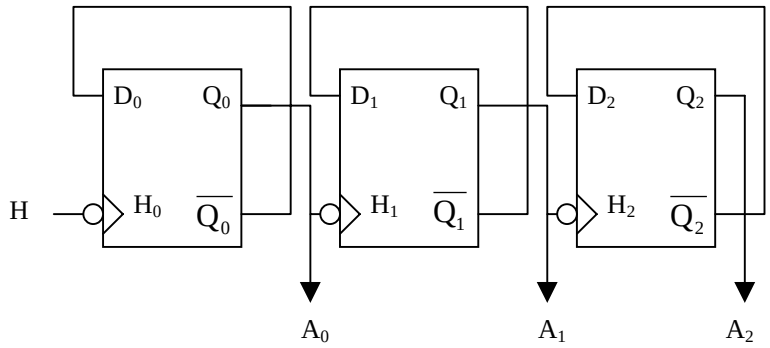
- *Table de fonctionnement :*

N	A_0	A_1	A_2
0	0	0	0
1	1	0	0
2	0	1	0
3	1	1	0
4	0	0	1
5	1	0	1
6	0	1	1
7	1	1	1
0	0	0	0

- *Chronogramme :*



- *Schéma de principe :*



Le nombre compté en sortie s'écrit : $N = A_2A_1A_0$.

Remarques :
Il faut que la **synchronisation** se fasse **sur** le **front descendant** de l'horloge. Sur front montant, on obtient un décompteur.

- Un tel compteur est utilisé pour la division de fréquence lors de la fabrication d'horloges.

4.3.1 Compteur synchrone

Ce compteur permet de **synthétiser n'importe quel code de façon systématique**.
L'horloge est distribuée sur toutes les entrées H et les configurations de sortie sont obtenues par un conditionnement des entrées au temps t_n en prévision du « coup » d'horloge t_{n+1}.

Dans un système logique synchrone, plusieurs considérations sont à prendre en compte :

- **la fréquence de l'horloge du système logique** est très souvent inférieure à la fréquence de référence dérivée d'un quartz. En effet, plus la fréquence de fonctionnement du système logique est élevée, plus la consommation est importante et plus l'échauffement de la puce est important. Ce point conduit à choisir, pour fréquence d'horloge du système logique, la fréquence minimale requise pour le sous-ensemble le plus exigeant au point de vue vitesse. Il y a lieu de signaler ici que la fréquence de pilotage d'un affichage multiplexé peut être quelconque par rapport à la fréquence de l'horloge du système logique. En effet, le signaux pilotant les afficheurs ne sont jamais rebouclés vers le système logique. Ils font partie de la périphérie de ce système tout comme les signaux de synchronisation.

- **Une division de fréquence asynchrone** est possible pour passer de la fréquence du quartz à celle de l'horloge, puisque le signal de l'horloge constitue la référence temporelle. La façon dont ce signal a été généré importe peu. Or un diviseur asynchrone nécessite beaucoup moins de composants qu'un diviseur synchrone : il est constitué de simples bascules D dont les entrées sont reliées aux sorties complémentées de ces bascules. En conséquence le diviseur consommera moins d'énergie et sera plus simple à concevoir.

Synthèse d'un compteur avec des bascules D

Cette technique consiste à **décrire** dans une table de vérité **la succession des états souhaités** puis **les valeurs que doivent prendre les entrées D** pour assurer cette succession d'états. Si après l'état 0, le compteur doit passer à l'état 1, il faut dès l'état 0 donner aux entrées D les valeurs de l'état 1. Ainsi au front actif suivant de l'horloge, les valeurs des entrées D seront transférées vers les sorties Q et le compteur se trouvera à l'état 1. Les sorties Q_i sont donc les fonctions d'entrée de la table de vérité tandis que les entrées des bascules D_i constituent les fonctions à synthétiser.
La technique de synthèse présentée dans ce cours consiste à établir les équations donnant les D_i en fonction des Q_i dans le cas général d'un compteur modulo 2^n, puis de modifier ces expressions en fonction du compteur modulo = 2^n à réaliser.

Equations donnant les D_i pour un compteur modulo 2^n
On procède par l'exemple puis on déduit les relations pour n quelconque.

Exemple : compteur modulo 2^3.

- *Table de vérité :*

N	Q_2	Q_1	Q_0	D_2	D_1	D_0
0	0	0	0	0	0	1
1	0	0	1	0	1	0

2	0	1	0	0	1	1
3	0	1	1	1	0	0
4	1	0	0	1	0	1
5	1	0	1	1	1	0
6	1	1	0	1	1	1
7	1	1	1	0	0	0
0	0	0	0	0	0	1

- *Equations donnant les D_i en fonction des Q_i :*

A partir de la table de vérité, on écrit :

$$D_0 = \overline{Q_0}$$

$$D_1 = \overline{Q_2}.\overline{Q_1}.Q_0 + \overline{Q_2}.Q_1.\overline{Q_0} + Q_2.\overline{Q_1}.Q_0 + Q_2.Q_1.\overline{Q_0} = \overline{Q_2}.\left(\overline{Q_1}.Q_0 + \overline{Q_1}.Q_0\right) + Q_2.\left(\overline{Q_1}.Q_0 + Q_1.\overline{Q_0}\right)$$

$$= \overline{Q_2}.\left(Q_1 \oplus Q_0\right) + Q_2.\left(Q_1 \oplus Q_0\right) = \left(\overline{Q_2} + Q_2\right).\left(Q_1 \oplus Q_0\right) = Q_1 \oplus Q_0$$

$$D_2 = \overline{Q_2}.Q_1.Q_0 + Q_2.\overline{Q_1}.\overline{Q_0} + Q_2.\overline{Q_1}.Q_0 + Q_2.Q_1.\overline{Q_0} = Q_2.\left(\overline{Q_1}.\overline{Q_0} + \overline{Q_1}.Q_0 + Q_1.\overline{Q_0}\right) + \overline{Q_2}.Q_1.Q_0$$

$$= Q_2.\overline{Q_1.Q_0} + \overline{Q_2}.Q_1.Q_0 = Q_2 \oplus Q_1.Q_0$$

- *Généralisation :*

En généralisant, on obtient : $D_0 = \overline{Q_0}$ et $D_i = Q_i \oplus \left(Q_{i-1}.Q_{i-2}...Q_0\right) = Q_i \oplus \left(\prod_{k=1}^{i} Q_{i-k}\right)$

Compteur modulo $N \neq 2^n$ - Méthode de conditionnement des entrées

Il faut, à l'état N-1, remettre à 0 les équations qui ne le sont pas, c'est à dire agir sur les entrées des bascules qui ne seront pas au niveau logique 0 à l'état N.

Nous expliquons ce principe pour un compteur modulo 44.

Un tel compteur nécessite 6 bascules (2^6=64>44). Nous déterminons les bascules sur lesquelles il faudra agir à l'état 43 par la fonction de décodage s43.

	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
	32	16	8	4	2	1
Etat 43 (s43 = 1)	1	0	1	0	1	1
Etat 44 (s44 = 1)	1	0	1	1	0	0

En explicitant l'état 44 du compteur, on s'aperçoit que les bascules Q_0, Q_1 et Q_4 seront à l'état 0 après l'état 43, ce que nous souhaitons ; il n'y a donc pas lieu de modifier leurs équations. Par contre, il faut remettre à 0 les entrées D_2, D_3 et D_5. Nous le ferons en multipliant leur équation respective par $\overline{s43}$, terme qui a la valeur 0 à l'état 43. La fonction de décodage s43 a pour équation :

$$s43 = Q_5.\overline{Q_4}.Q_3.\overline{Q_2}.Q_1.Q_0$$

En utilisant le principe de la première coïncidence, nous faisons disparaître les termes complémentés car cette coïncidence de 1 pour Q_5, Q_3, Q_1 et Q_0 ne se retrouve pas dans le cycle de comptage avant l'état s43 (ceci est une procédure systématique toujours applicable). D'où les équations définitives :

$$s43 = Q_5.Q_3.Q_1.Q_0$$

$$D_0 = \overline{Q_0}$$

$$D_1 = Q_1 \oplus Q_0$$

$$D_2 = \left[Q_2 \oplus \left(Q_1.Q_0\right)\right].\overline{s43}$$

$$D_3 = \left[Q_3 \oplus \left(Q_2.Q_1.Q_0\right)\right].\overline{s43}$$

$$D_4 = \left[Q_4 \oplus \left(Q_3.Q_2.Q_1.Q_0\right)\right]$$

$$D_5 = \left[Q_5 \oplus \left(Q_4.Q_3.Q_2.Q_1.Q_0\right)\right].\overline{s43}$$

Remarque : la remise à 0 du compteur peut aussi se faire via les entrées de forçage (CLR) des bascules, il faut pour le cas précédent, activer les entrées de forçage CLR avec la fonction $\overline{s44}$.

Dans ce cas, le problème est qu'il existe un état transitoire non souhaité en sortie du système.