

Éléments de conception des systèmes numériques

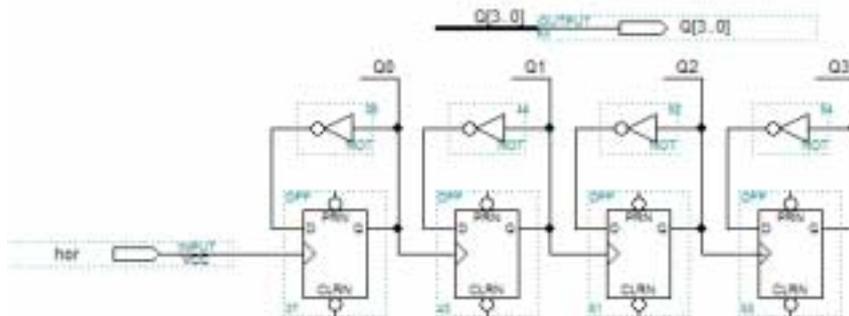
L'évolution des circuits intégrés numériques, en particuliers les circuits configurables (CPLD, FPGA), conduit le concepteur à réaliser des systèmes de plus en plus complexe, où un défaut devient très difficile à mettre en évidence par la simulation, et encore plus difficile à localiser. Afin d'éviter de longues heures de recherche d'une erreur, il convient de mettre en œuvre quelques règles de conception.

1 Système synchrone ou asynchrone

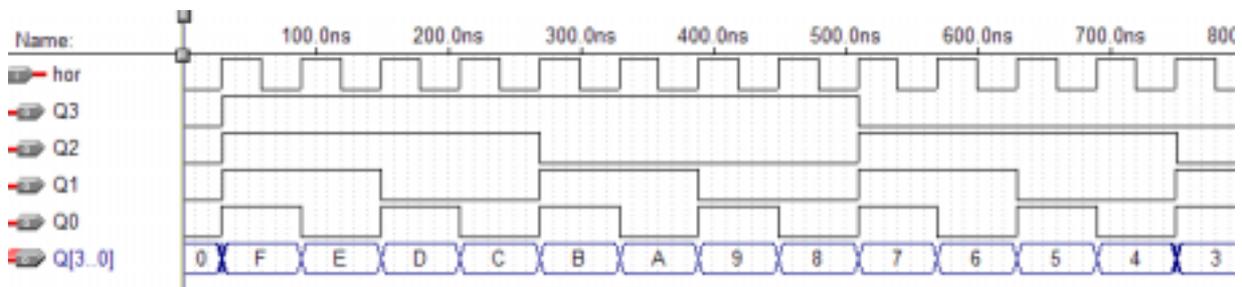
Une des première question à se poser est lié au choix d'un système asynchrone ou synchrone. Dans le premier les bascules peuvent changer d'état à n'importe quel moment en fonction du signal arrivant sur leur entrée d'horloge, dans le second ces entrées sont reliées à une horloge commune et les changement d'état se font tous au même moment ; le système est alors cadencé uniquement par cette horloge.

1.1 Exemple de système asynchrone

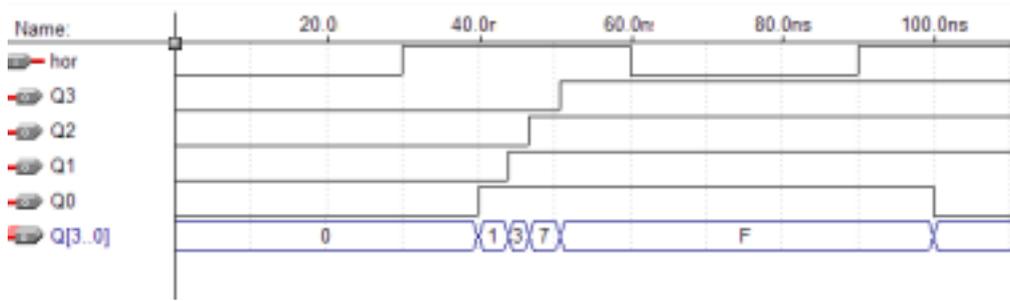
Pour mettre en évidence les problème liés à l'asynchronisme d'un systèmes, prenons le classique exemple du décompteur (plus simple à réaliser que le compteur). La première réalisation proposée est un système asynchrone : l'entrée horloge des bascules utilisées n'est donc par définition pas reliés à une horloge commune.



Une simulation purement fonctionnelle donne bien le résultat attendu :



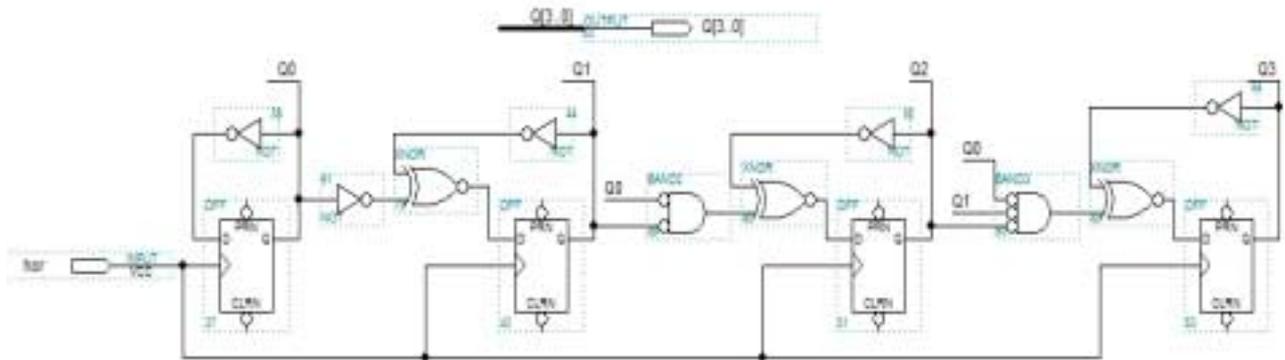
Avec une simulation prenant en compte le temps de retard des bascules, pour une fréquence d'horloge suffisamment élevée, on observe l'apparition d'états indésirables : par exemple lors de la transition de l'état 0 vers l'état F, il faut passer par les états 1 puis 3 puis 7 : en effet, c'est le changement d'état d'une bascule qui provoque le changement d'état de la suivante et ainsi de suite.



Cette situation, sans conséquence si on souhaite commander un afficheur (l'œil ne sera pas assez rapide par rapport à la vitesse de transition des bascules pour voir les états intermédiaires), devient inacceptable si ce décompteur s'inclut dans un ensemble rapide, qui risque de lire de valeurs erronées si aucune précaution n'est prise.

1.2 Exemple de système synchrone

Pour pallier le problème précédent, une seule solution : faire en sorte que les entrées d'horloge des bascules soit communes ; le système devient alors synchrone. Il s'ensuit une complexité plus importante comme le montre le schéma ci-dessous, qui réalise la même fonction que le précédent.

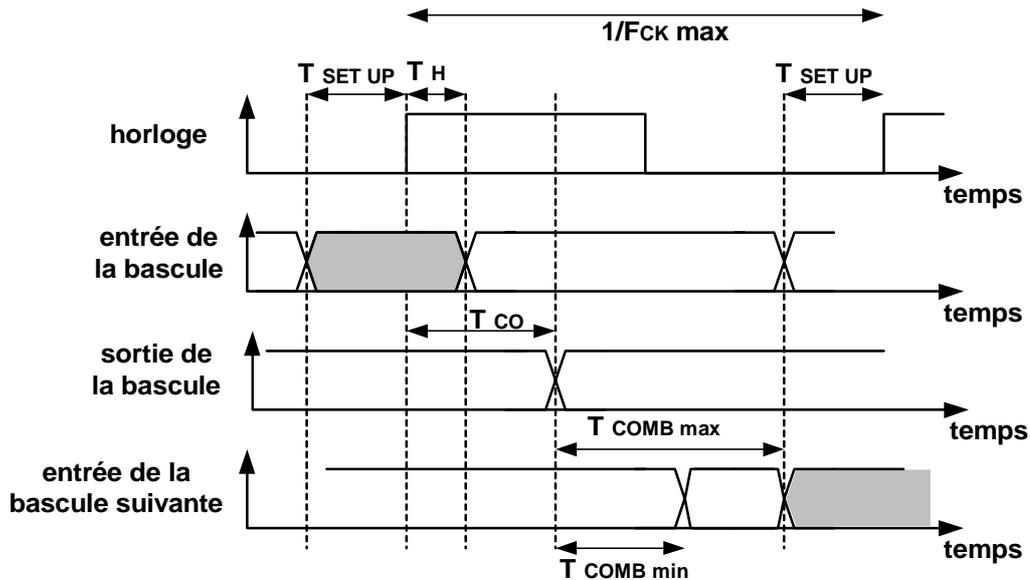


Une simulation temporelle montre alors que les bascules changent d'état toutes en même temps, un temps de retard T_{CO} (temps de propagation « clock » vers « output ») après le front actif d'horloge.



1.3 Comparaison des deux systèmes

Un des critères de choix sera la fréquence maximale admissible pour l'horloge. Le signal sur l'entrée D des bascules doit être stable un temps T_{SETUP} (temps de pré-positionnement) avant le front d'horloge. En sortie Q des bascules, le signal apparaît un temps T_{CO} (temps de propagation « clock » vers « output ») après le front actif. Le signal est disponible sur l'entrée D de la bascule suivante après un temps T_{COMB} correspondant au passage de la couche combinatoire. Le nouveau front d'horloge peut alors apparaître après le temps de pré-positionnement. Sur le chronogramme suivant, la zone où les entrées de bascules doivent rester stable a été grisée. La zone hachurée représente l'incertitude sur T_{COMB} (voir paragraphe suivant).



On en déduit donc que la fréquence d'horloge F_{CK} doit vérifier la relation :

$$F_{CK} \leq \frac{1}{T_{SETUP} + T_{CO\ max} + T_{COMB\ max}}$$

Dans le cas d'un système asynchrone, cette expression est à considérer indépendamment pour chaque bascule, tandis que pour le système synchrone elle est valable pour l'ensemble des bascules si on suppose des temps T_{COMB} identiques.

Les temps T_{SETUP} et T_{CO} dépendent du circuit choisi, tandis que T_{COMB} dépend de la complexité de la conception.

Un système asynchrone étant moins complexe, permet donc une vitesse d'horloge plus importante, tandis que le système synchrone assurera la fiabilité indispensable à un système rapide. C'est pour cette dernière raison que les conceptions nouvelles sont généralement synchrones.

Dans le cas particulier des deux montages précédents, on utilisera le montage asynchrone pour réaliser un diviseur de fréquence si une seule sortie sera utilisée. On profitera ainsi de la simplicité et de la rapidité de ce premier montage. Pour réaliser un compteur, on utilisera plutôt le montage synchrone, évitant ainsi tout état intermédiaire. La lecture de la sortie se fera également par un registre synchrone (avec la même horloge que le compteur), ce qui permettra de s'affranchir des temps de retard.

Remarques : pour un fonctionnement correct, on doit également vérifier que la donnée à l'entrée de la bascule D reste présente un temps T_H (temps de maintien) après le front actif d'horloge. Le signal variera au bout d'un temps correspondant à $T_{CO} + T_{COMB}$, d'où :

$$T_H < T_{CO\ min} + T_{COMB\ min}$$

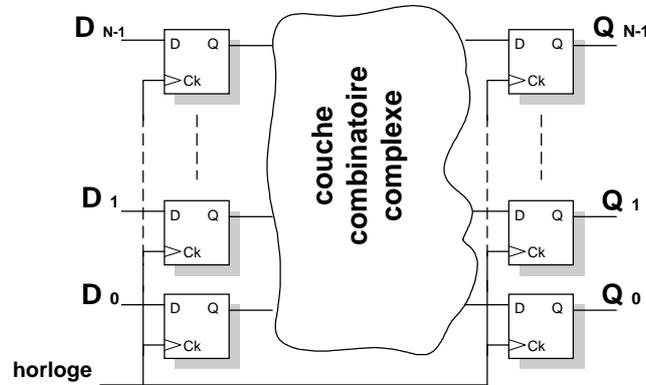
Cette relation ne pose pas de problème, le temps de maintien étant souvent nul ou très faible.

2 Problèmes et solutions liés aux montages synchrones

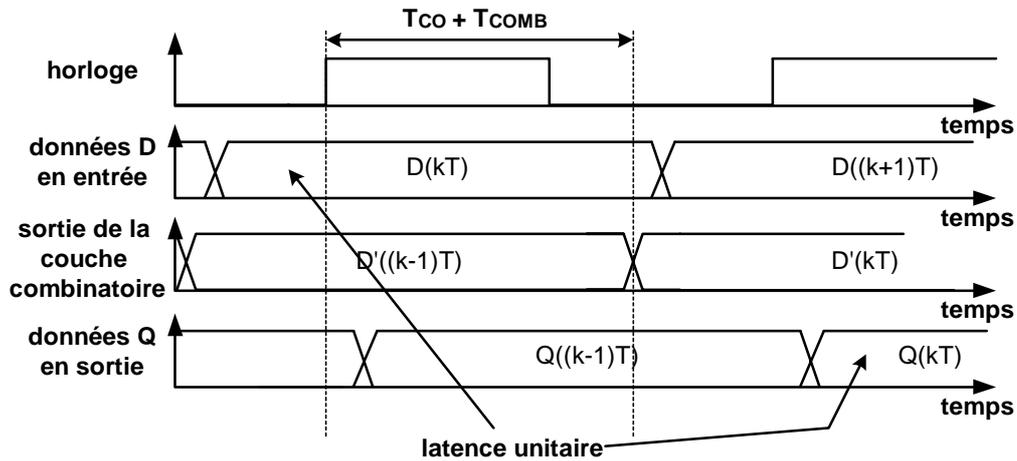
2.1 Montage « pipeline » et temps de latence

Comme nous l'avons vu, la vitesse de fonctionnement d'un système synchrone est essentiellement limitée par la complexité de la couche combinatoire : plus celle-ci sera complexe, plus T_{COMB} sera important. Il convient de noter également que ce temps n'est pas prédictible dans le cas de l'utilisation d'un FPGA (il dépend du routage), et une fois le routage effectué, il dépend de la donnée présente à l'entrée de la couche combinatoire. Il devient donc très compliqué de prévoir la vitesse maximale de fonctionnement si la couche combinatoire est importante. La solution couramment utilisée consiste à fractionner la partie combinatoire entre plusieurs bascules cadencées par l'horloge ; on parle alors de système en « pipeline ».

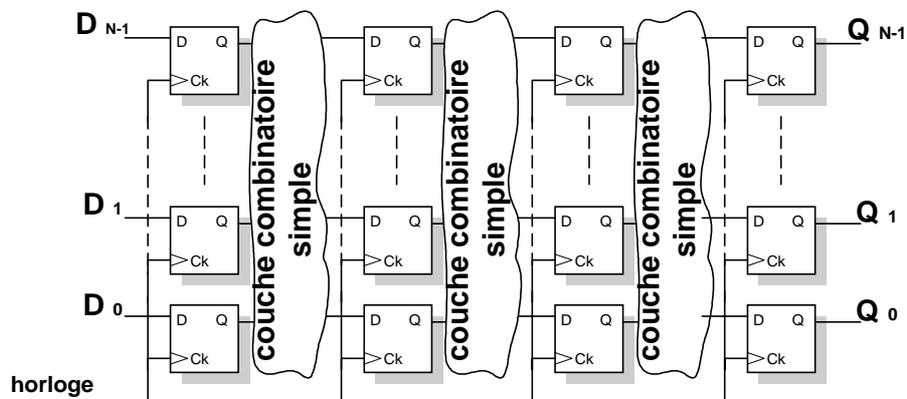
Illustrons ce propos par un exemple, un peu plus compliqué que notre compteur, qui n'avait pas d'entrées de données. Imaginons que nous voulions faire l'acquisition d'une donnée D sur N bits, pour la multiplier par une constante, et obtenir en sortie une donnée Q re-synchronisée de nouveau sur l'horloge. La multiplication par la constante se traduira par une couche combinatoire plus ou moins importante suivant le nombre de bits N :



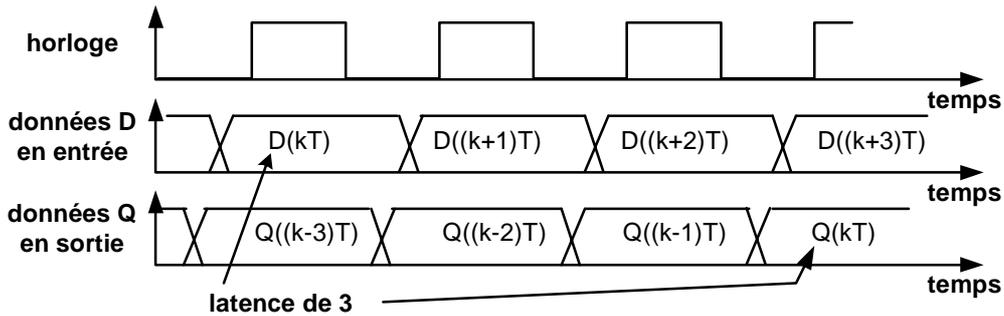
Les chronogrammes sont alors les suivants, où $D(kT)$ représente la donnée D à l'instant kT :



Pour augmenter la vitesse de fonctionnement, on va fractionner la couche combinatoire en plusieurs sous-couches entre lesquelles on intercale des bascules :



Le temps T_{COMB} est alors diminué, la fréquence maximale d'horloge et le débit de données peuvent alors être augmentés. En contre partie, il existe un temps, dit temps de latence, de X coups d'horloge (trois dans notre exemple), entre le moment où une donnée entre et le moment où on la retrouve en sortie.

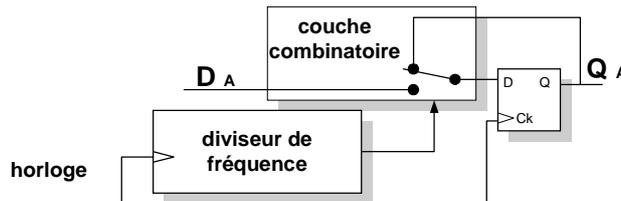


2.2 Montage à plusieurs fréquences d'acquisition

Il est courant d'avoir à faire l'acquisition d'une donnée D_A , tout les K coups d'horloge. La solution qui consisterait à diviser l'horloge principale par K , pour commander la ou les bascules d'acquisition, conduit finalement à un asynchronisme, du au temps de retard introduit par le diviseur.



Le montage suivant, où les horloges sont communes, mais où la ou les bascules d'acquisition de D_A lisent D_A une fois sur K coups d'horloges, et leur propre sortie les $K-1$ coups restants, permet de résoudre le problème.



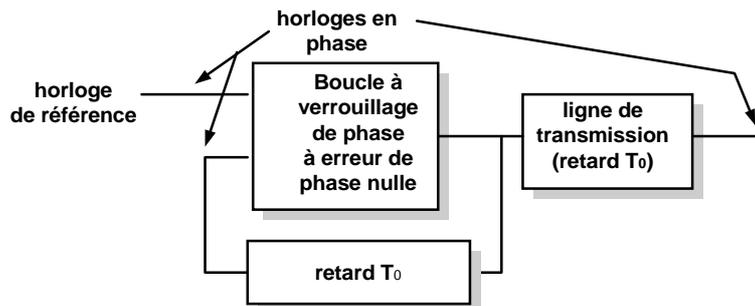
2.3 Distribution de l'horloge et « skew »

Que ce soit au sein d'une carte ou à l'intérieur d'un FPGA, la distribution des signaux d'horloge est délicate. En effet tout retard de propagation conduit à des fronts d'horloge n'arrivant pas au même moment sur les différentes bascules, et donc à un asynchronisme. Ce phénomène est désigné par le terme anglo-saxon de « skew », dont la valeur classique est de quelques centaines de ps sur une carte imprimée.

Les principales causes sont les temps de propagation d'un signal sur un ligne (capacité et inductance parasites de la ligne), et pour les FPGA le passage des points de connexion (qui introduisent une résistance, interagissant avec les capacités parasites).

Dans le cas de FPGA, on limite le phénomène en distribuant l'horloge sur une ligne à part.

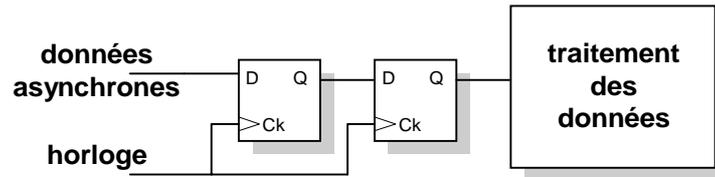
Dans tous les cas, on peut aussi utiliser une boucle à verrouillage de phase, à erreur de phase nulle, dont on a placé dans la boucle de contre réaction, un temps de retard égal à celui de la propagation sur la ligne. Les signaux à l'entrée du comparateur étant en phase, celui en sortie de la boucle sera donc en avance du temps de retard, compensant celui-ci.



2.4 Méta stabilité et acquisition asynchrone

Il arrive également qu'un système synchrone doive faire l'acquisition d'une donnée asynchrone, par exemple lorsque deux systèmes numériques communiquent sans transmission séparée du signal d'horloge. Si la donnée que l'on souhaite acquérir change sans respecter le temps de pré-positionnement ou le temps de maintien de la bascule d'acquisition, celle-ci risque de passer dans un état « métastable ». Le constructeur des composants ne garantit alors plus le fonctionnement, la donnée acquise pouvant être correcte ou erronée, ou bien encore la bascule pouvant entrer en oscillation pour un temps indéterminé.

Il est impossible d'éviter complètement cet état de chose, on peut éventuellement faire un calcul statistique sur la probabilité d'apparition du phénomène, calcul sortant du cadre de cette exposé. Ce calcul montre que la probabilité de propagation de l'oscillation est réduite si l'acquisition est faite par deux bascules :



Bibliographie

Logique programmable par L. Dutrieux et D. Demigny chez Eyrolles