
Bonnes pratiques de la conception pour FPGA



Pierre Langlois

<http://creativecommons.org/licenses/by-nc-sa/2.5/ca/>

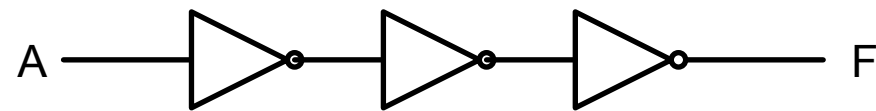
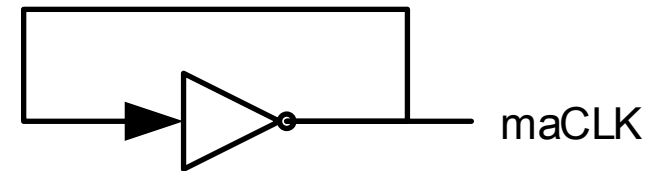
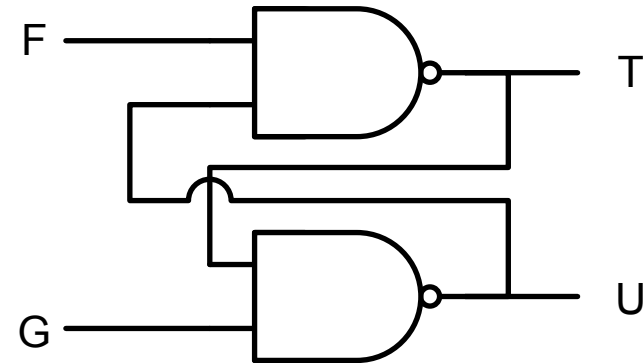
Sept principes de conception FPGA

- Sept principes de conception applicables principalement aux implémentations sur FPGA.
 1. Adopter un style de conception synchrone;
 2. Utiliser des bascules plutôt que des loquets;
 3. Définir un signal d'initialisation globale;
 4. Utiliser les réseaux de distribution de signaux d'horloge;
 5. Exploiter toutes les ressources disponibles sur le FPGA;
 6. Effectuer une disposition manuelle des blocs sur le chemin critique; et,
 7. Favoriser l'arithmétique en virgule fixe plutôt que flottante.

Principe de conception FPGA #1:

Conception synchrone

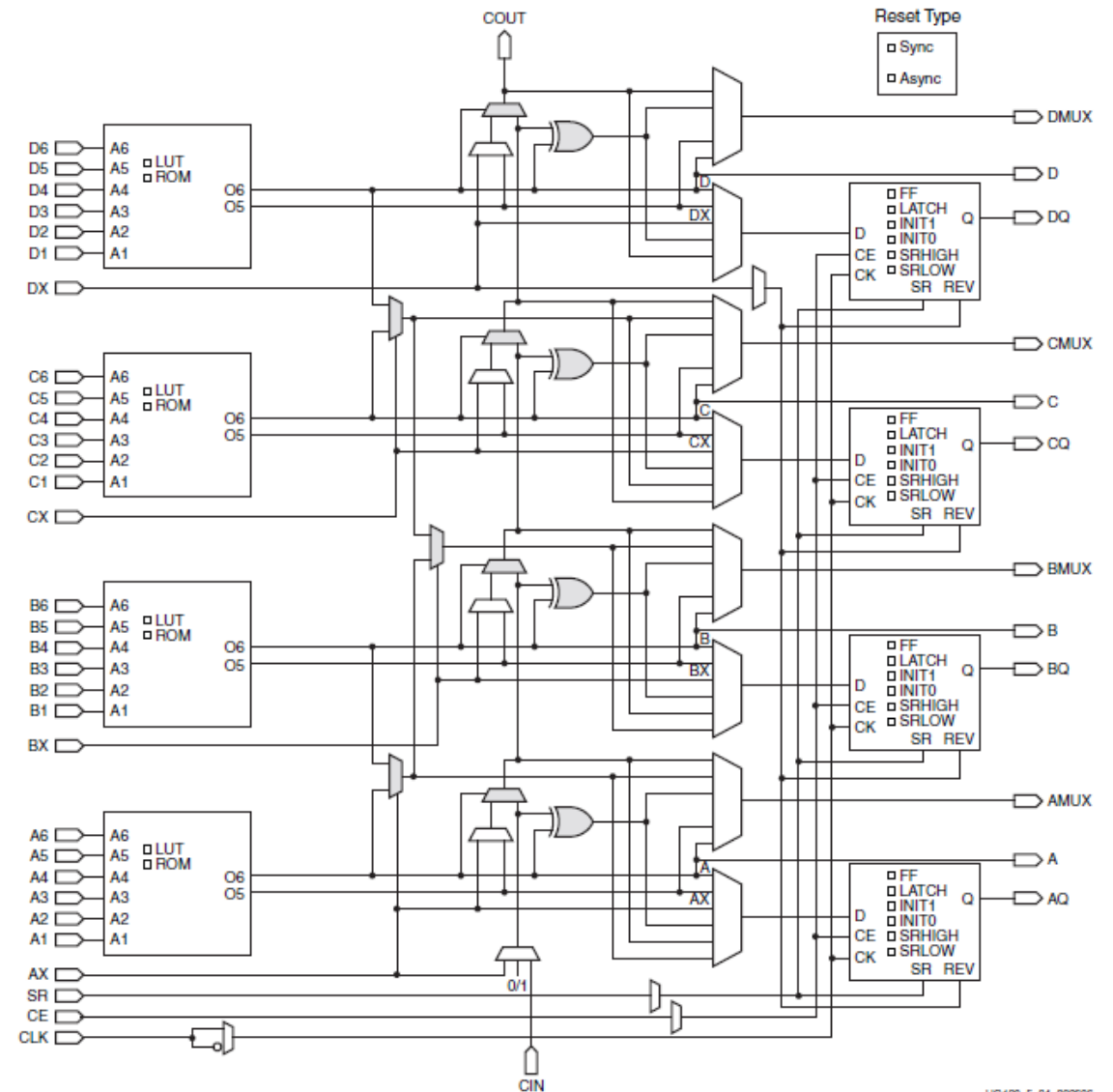
- Les structures asynchrones sont mal adaptées aux FPGA. On obtient des résultats fiables avec un FPGA en adoptant un style de conception synchrone.
- Il faut éviter de se fier aux délais d'une composante ou du routage d'un signal, surtout si on utilise un synthétiseur et des outils d'implémentation automatiques. Par exemple, on pourrait être tenté de retarder un signal en le faisant passer par deux inverseurs en cascade. Cependant, un synthétiseur aura tôt fait d'éliminer cette structure inutile du point de vue fonctionnel.
- Les signaux d'initialisation des bascules devraient aussi être synchrones.



Principe de conception FPGA #2:

Bascules vs loquets

- Dans un ASIC sur mesure ou un ASIC à cellules normalisées, les loquets offrent plusieurs avantages. Tout d'abord, ils requièrent la moitié des transistors d'une bascule. En utilisant des techniques spéciales, on parvient à obtenir des circuits fiables et plus rapides qu'avec des bascules.
- Pour un FPGA, le coût est identique entre une bascule et un loquet. Il est donc grandement avantageux de toujours utiliser des bascules.



Principe de conception FPGA #3:

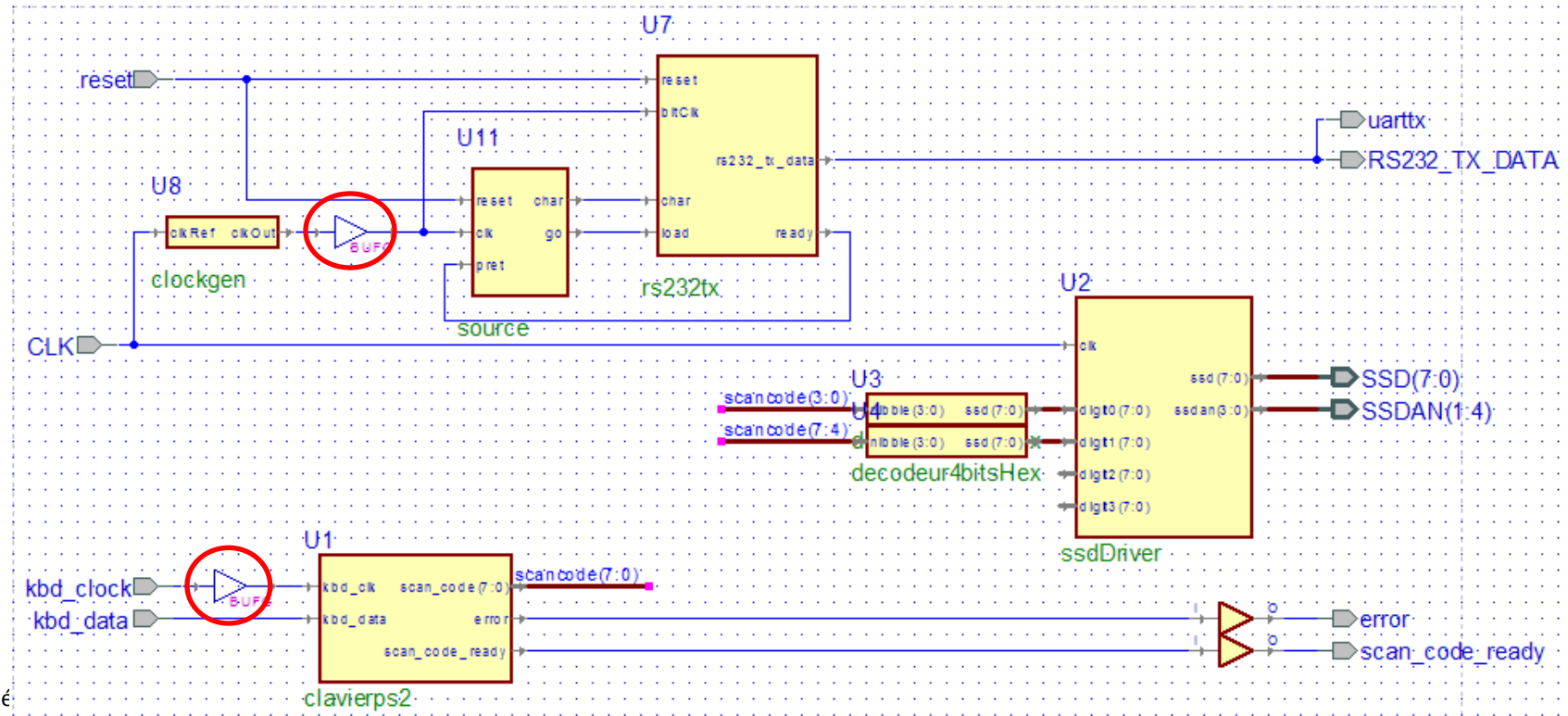
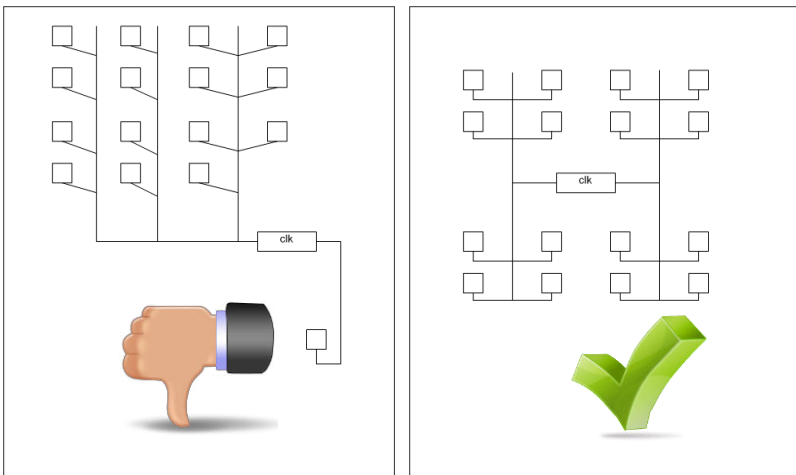
Initialisation globale

- Il est avantageux d'utiliser un signal unique pour initialiser toutes les bascules parce que la plupart des FPGA comportent un chemin spécial pour accomplir cette tâche.

```
process (CLK, reset)
begin
    if rising_edge(CLK) then
        if reset = '1' then
            lesRegistres <= (others => (others => '0'));
        else
            ...
        end if;
    end process;
```

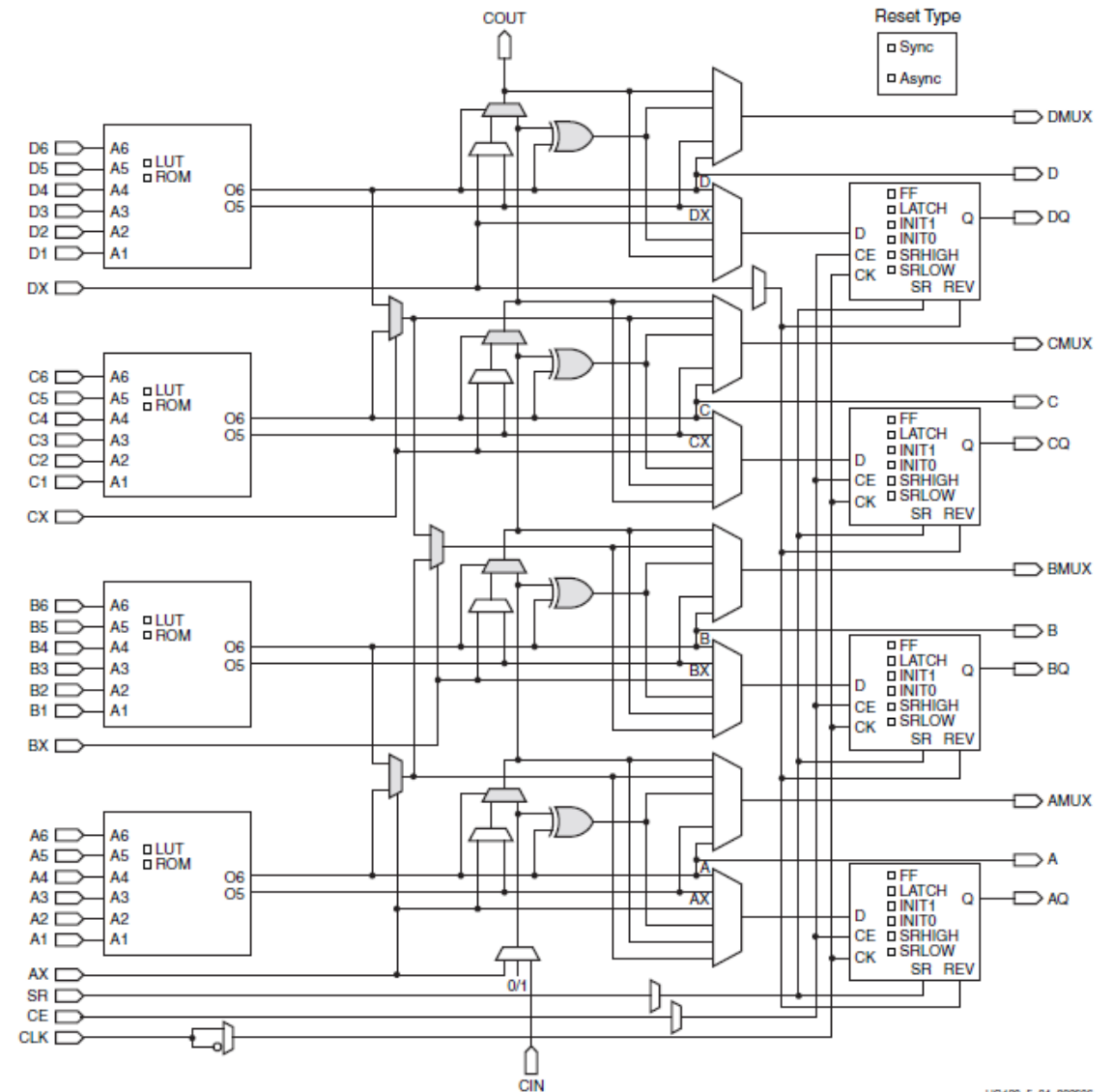
Principe de conception FPGA #4: Génération et distribution de l'horloge

- Le routage inadéquat d'un signal d'horloge peut résulter en un déphasage d'horloge et une réduction des performances du circuit.
- Un signal d'horloge généré sur le FPGA devrait être passé à un tampon d'horloge (Xilinx: Global Clock Buffer, un Virtex-5 en contient 32). On insère le symbole « BUFG » à la sortie du générateur d'horloge.



Principe de conception FPGA #5: Exploitation de toutes les ressources (1)

- Les ressources en place dans un FPGA sont 'gratuites'.
 - Une fois une puce choisie pour un système, il ne coûte 'rien' d'utiliser toutes les ressources qu'elle contient.
 - Ce raisonnement n'est pas vrai nécessairement pour les autres technologies, comme pour les ASICs à cellules normalisées.
- Il est très avantageux d'utiliser les registres des tranches pour pipeliner les chemins de données.
 - Augmenter le débit (mais la latence aussi).
 - Plusieurs applications sont insensibles à une latence de plusieurs dizaines de cycles d'horloge de 100 MHz ou plus.



Principe de conception FPGA #5: Exploitation de toutes les ressources (2)

- Un corolaire de ce principe est qu'il est avantageux de placer deux registres supplémentaires dans le chemin de chaque entrée et sortie du système.
- Le premier registre est placé dans le bloc d'entrées-sorties du FPGA, diminuant au minimum le délai pour la saisie du signal.
- Le deuxième registre donne beaucoup de flexibilité au placeur et au routeur pour déterminer l'emplacement optimal du module qui doit traiter ce signal.

Principe de conception FPGA #5:

Exploitation de toutes les ressources (3)

- Les délais de propagation peuvent être grandement réduits en limitant la charge à la sortie d'un module.
 - On peut atteindre ce but en employant un arbre de distribution.
 - Le principe consiste à faire mener une charge réduite composée d'un groupe de bascules.
 - Chacune de ces bascules peut ensuite mener les modules qui doivent recevoir le signal.
 - Dans la description du circuit, il s'agit d'utiliser des noms différents pour des signaux identiques, ce qui devrait empêcher l'outil de synthèse de les combiner.

Principe de conception FPGA #6:

Disposition manuelle

- Il peut être très avantageux de faire la gestion manuelle de la disposition (*floorplanning*) pour les parties critiques d'un circuit.
- Les fabricants de FPGA offrent des outils qui permettent de déterminer la position absolue ou relative de certains registres sur la puce.
- On laisse ensuite les outils automatisés faire la disposition du reste du circuit.
- En plaçant manuellement les blocs les plus lents, on facilite la tâche du placeur automatique et on accélère cette étape.
- L'avantage principal, cependant, est qu'on peut en arriver à une solution réellement optimale (i.e. qui ne peut être améliorée).
- L'inconvénient est que ce travail peut être ardu et pénible, et il devrait donc être réservé uniquement aux parties critiques d'un système.

Principe de conception FPGA #7:

Arithmétique en virgule fixe

- On requiert rarement la très grande précision (8 chiffres décimaux significatifs et plus) offerte par les microprocesseurs à usage général.
- Ceci est spécialement vrai pour les systèmes qui traitent des données provenant de phénomènes naturels comme les sons ou les signaux électromagnétiques.
 - La voix dans un système téléphonique est échantillonnée avec 8 bits.
 - Les sons enregistrés sur un CD audio le sont avec 16 bits.
 - Les pixels sont encodés sur 8 bits.
- Il y a au moins un ordre de grandeur de différence en complexité et en performance entre un circuit arithmétique qui fonctionne en virgule fixe par rapport à un circuit semblable qui fonctionne en virgule flottante.
- Il est utile lors de la modélisation du circuit de tenir compte de ce fait et de mesurer les conséquences d'utiliser un système à virgule fixe.
 - 10 bits donnent une précision de 3 chiffres significatifs.
 - 20 bits donnent une précision de 6 chiffres significatifs.

Vous devriez maintenant être capable de ...

- Appliquer les sept principes suivants pour guider l'utilisation de FPGA: conception synchrone, utiliser des bascules plutôt que des loquets, initialisation globale, distribution d'horloge, utilisation des ressources 'gratuites', disposition manuelle et calculs en virgule fixe. (B3)

Code	Niveau (http://fr.wikipedia.org/wiki/Taxonomie_de_Bloom)
B1	Connaissance – mémoriser de l'information.
B2	Compréhension – interpréter l'information.
B3	Application – confronter les connaissances à des cas pratiques simples.
B4	Analyse – décomposer un problème, cas pratiques plus complexes.
B5	Synthèse – expression personnelle, cas pratiques plus complexes.