

TP2-3 : Convertisseur DA (8 bits) à transfert de charge

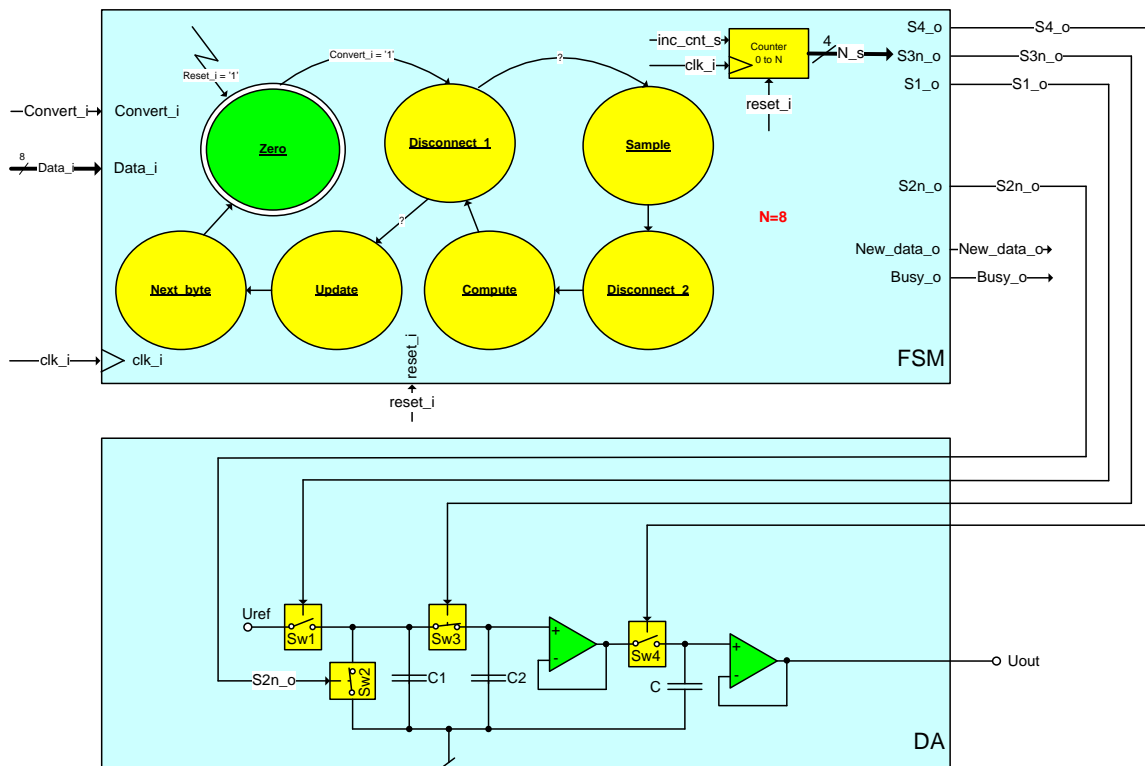
1. Buts du travail

- ✓ **Etudier et comprendre** le convertisseur DA à transfert de charge.
- ✓ **Coder en VHDL la machine d'état** nécessaire à la commande de ce circuit.
- ✓ **Réaliser et vérifier** le circuit complet (un générateur de sinus).

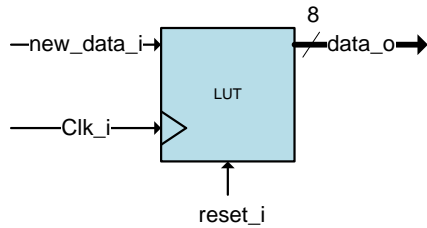
2 Description du circuit

Cette expérience de laboratoire illustre les possibilités des circuits à capacités commutées. La conversion se fait de manière séquentielle à raison de un cycle par bit. En fin de conversion, la valeur analogique est conservée par un échantillonneur-bloqueur pour permettre de décharger les condensateurs du D/A.

A chaque cycle, le condensateur d'évacuation C1 est chargé par Uref (si bit à 1) ou GND (si bit à 0) durant la période de l'horloge. Lors de la période inactive, la charge du condensateur d'évaluation est partagée avec celle du condensateur de sommation C2 : il y a égalisation des tensions par la fermeture de S3. Ainsi, à chaque cycle, l'importance de la charge précédente sur C2 diminue d'un rapport $\frac{1}{2}$. L'évaluation se fait donc du LSB au MSB.



LUT pour générateur de sinus



Entités:

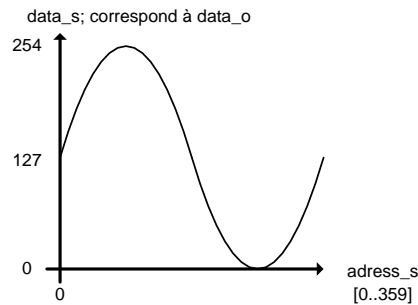
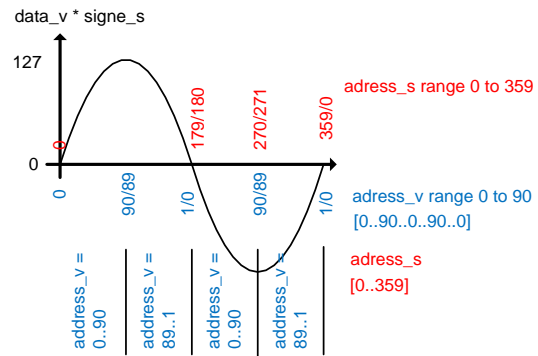
new_data_i, clk_i, reset_i : std_logic
data_o : std_logic_vector(7 downto 0)

Signaux internes:

signe_s : integer range -1 to 1
→ +1: adress_s de 0 à 180
→ -1: adress_s de 181 à 359
data_s : integer range 0 to 254
adress_s : integer range 0 to 359

Variables:

adress_v : integer range 0 to 90
data_v : integer range -127 to 127



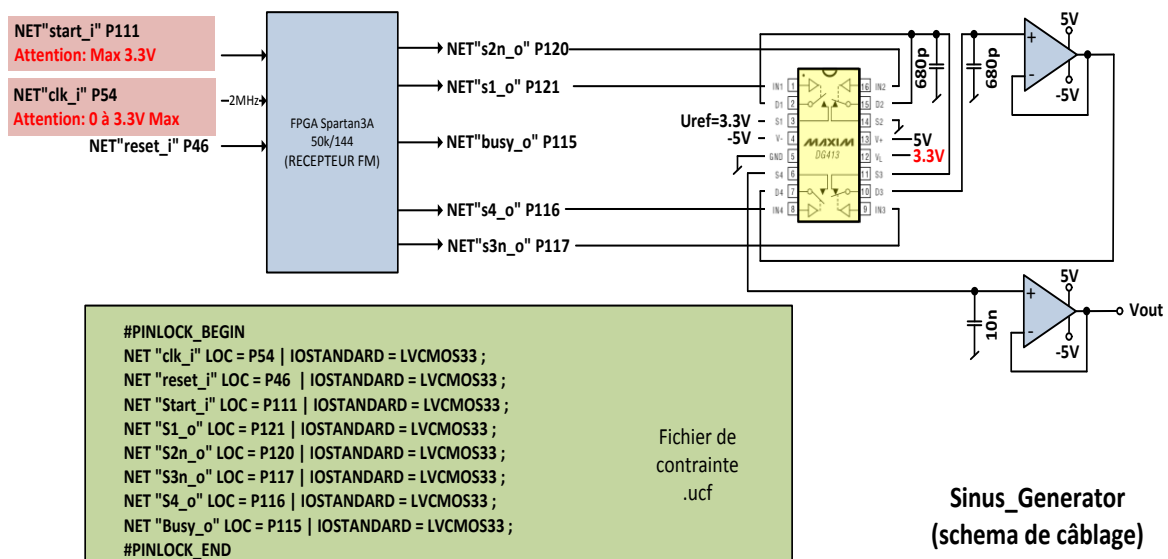
Description du fonctionnement:

Si l'entrée new_data_i vaut 1 alors, à chaque flanc montant de l'horloge, adress_s est incrémenté:

- $data_v * signe_s \leq 127 * \sin(adress_s)$
- $data_s \leq 127 + (data_v * signe_s)$

Si l'entrée new_data_i vaut 0 alors data_o garde son ancienne valeur. Si l'entrée asynchrone reset_i vaut 1, alors data_o s'initialise à la valeur 127.

Schéma pour le montage électronique



3 Travail à effectuer

Vous devez :

1. Écrivez le code VHDL pour la LUT permettant de générer le sinus
2. Vérifiez le bon fonctionnement de votre modèle de LUT avec la macro
3. Étudiez le fonctionnement du convertisseur D/A à l'aide du chronogramme en annexe
4. Complétez l'exercice concernant le convertisseur D/A
5. Définissez la machine d'état permettant de contrôler les interrupteurs (S1 à S4)
6. Réalisez le « Top-level » du système permettant de réaliser le convertisseur D/A à génération de sinus et vérifiez son bon fonctionnement avec le fichier .do
7. Réalisez le circuit analogique avec des AO, le circuit des commutateurs DG413 et des condensateurs
8. Synthétisez le code et implémentez-le dans la FPGA Spartan 3A de Xilinx
9. Mesurez le sinus à 8 bits que vous venez de réaliser à l'aide d'un oscilloscope
10. Répondez à la question : que faudrait-il faire si je voulais avoir la possibilité de varier la fréquence du sinus généré?

4 Information supplémentaire

Fonctionnement : Analysez le fonctionnement du D/A définissez le chronogramme complet des signaux de commande des quatre interrupteurs (S1 à S4).

Attention, notez que:

- Au début de chaque conversion, il faut décharger C1 et C2.
- Pendant la conversion D/A, pour éviter des décharges ou des court-circuits indésirés, veillez à toujours ouvrir tous les interrupteurs avant de les fermer.
- Pour l'analog switch « DG413 », S1 et S4 sont fermés avec une commande à 1 alors que S2 et S3 sont fermés avec une commande à 0 (voir datasheet)

Développer sur papier la machine d'état permettant de générer les sorties s1_o, s2n_o, s3n_o, s4_o, busy_o et new_data_o à partir des entrées clk_i, reset_i, convert_i et data_i (8 bits).

5 Rapport

Résumer vos travaux de développement et de vérification dans un **rapport de laboratoire**.

Annexes

