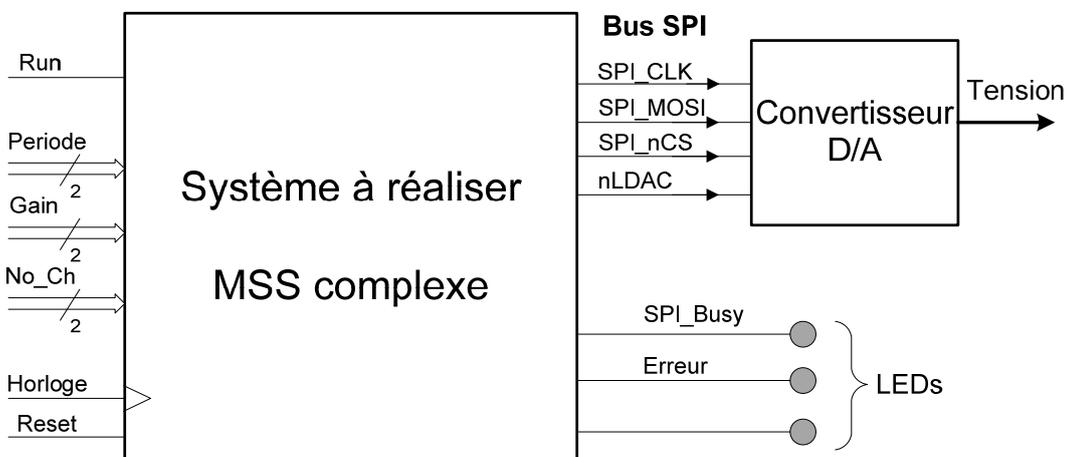


Convertisseur D/A sur ligne série SPI

Mandat

Le but de ce projet est de concevoir, réaliser et tester une description VHDL synthétisable d'un générateur de fonction comprenant une interface maître SPI. L'objectif est d'utiliser le convertisseur numérique/analogique disponible sur la carte REPTAR pour réaliser un générateur de fonction. Ce convertisseur dispose d'une interface série de type SPI. Le système sera conçu selon la méthodologie d'une machine séquentielle complexe. Il doit répondre aux spécifications données ci-après.

Schéma bloc du système:



Entrées

- *Reset_i* signal d'initialisation à action asynchrone.
- *Horloge_i* signal d'horloge du système d'une fréquence de 100 MHz.
- *Run_i* entrée activant le fonctionnement du générateur de fonction.
- *No_Ch_i* numéro du canal du convertisseur, sélection de 0 à 3, signal de 2 bits.
- *Periode_i* sélection de la période du signal généré, signal de 2 bits.
- *Gain_i* choix de l'amplitude (gain) de la tension de sortie, signal de 2 bits.

Sorties

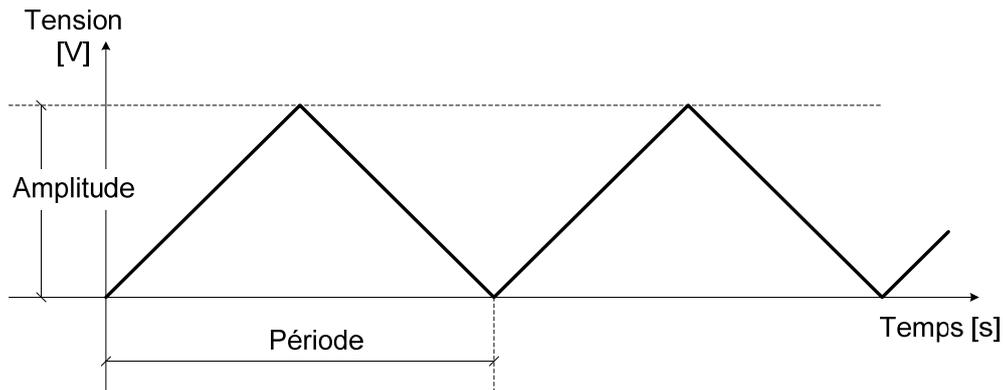
Signaux du bus série synchrone SPI

- *SPI_CLK_o* signal fournissant l'horloge du bus série SPI
- *SPI_MOSI_o* signal "Master Output, Slave Input", transfert en série des données du maître vers l'esclave.
- *SPI_nCS_o* signal de sélection de l'esclave, signal actif bas
- *nLDAC_o* signal de mise à jour simultané de tous les convertisseurs avec les valeurs reçues en série par l'interface SPI, signal actif bas.

- Erreur_o signal indiquant une erreur du fonctionnement du maître SPI
- SPI_Busy_o signal indiquant que le maître SPI est occupé, un transfert est en cours.
- à définir tous autres signaux permettant le dépannage du système

Réalisation d'un générateur de fonction

Nous souhaitons réaliser un générateur de fonction triangulaire avec le convertisseur D/A piloté via la ligne série SPI. La période ainsi que l'amplitude du signal triangulaire généré seront modifiable. La figure ci-dessous montre l'allure de la tension à générer:



La valeur du gain permet de choisir 4 amplitudes de la tension délivrée par le convertisseur D/A, soit:

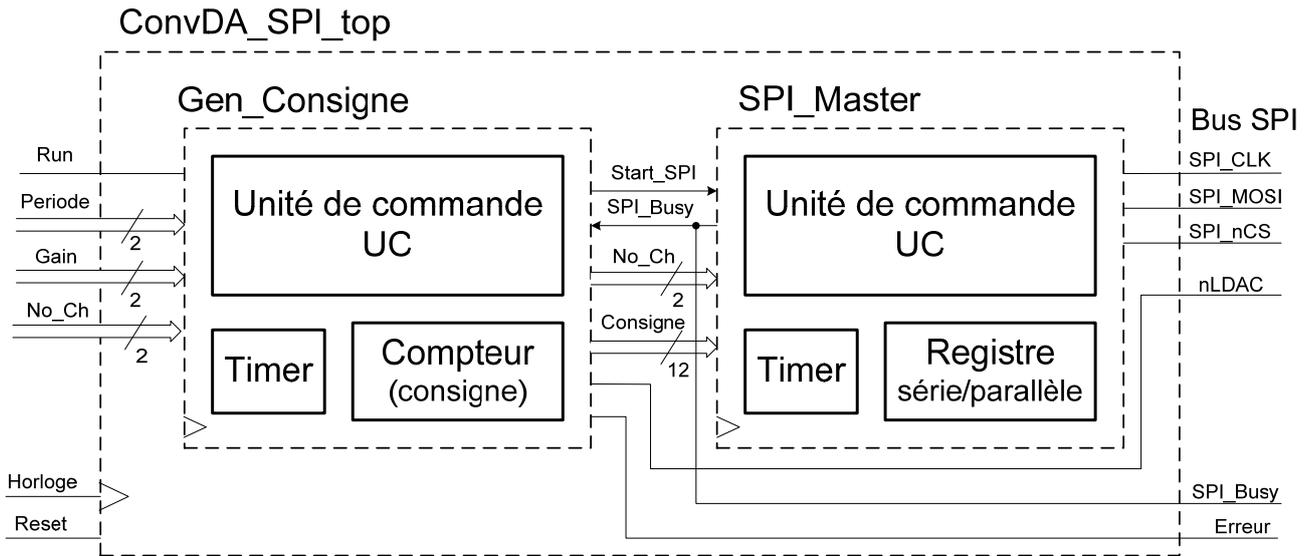
| Gain (2 bits) | Amplitude de la tension |
|---------------|-------------------------|
| 00 | 25% |
| 01 | 50% |
| 10 | 75% |
| 11 | 100% |

Le code de la période permet de choisir 4 fréquences pour la fonction triangle générée par le convertisseur D/A, soit:

| Période (2 bits) | Fréquence du signal généré |
|------------------|----------------------------|
| 00 | 50 Hz |
| 01 | 100 HZ |
| 10 | 200 Hz |
| 11 | 400 Hz |

Décomposition du système

La complexité du système est trop importante pour une réalisation en une seule machine séquentielle. Nous pouvons identifier deux sous-machines, soit: un système qui calcul la consigne et un second qui gère le transfert de la consigne au convertisseur D/A via la ligne série SPI. Le générateur de consigne est constitué d'un compteur à pas variable permettant de générer les valeurs de consigne selon le gain et la période souhaitée. Le transfert série est assuré par un maître SPI (SPI-Master). Celui-ci est constitué d'un registre à décalage permettant de transmettre la consigne en série sur le bus SPI et d'un compteur pour générer l'horloge du bus à la fréquence adéquate. La figure ci-après donne la décomposition proposée pour ce projet:



Spécification du fonctionnement souhaité

Lorsque le signal Run est actif, le système doit générer et transmettre les valeurs de tension successives pour réaliser la fonction triangle souhaitée. La succession des valeurs de consignes doivent être calculées de façon à réaliser une fonction triangle qui a l'amplitude et la période définie en entrée du système (signaux Periode et Gain).

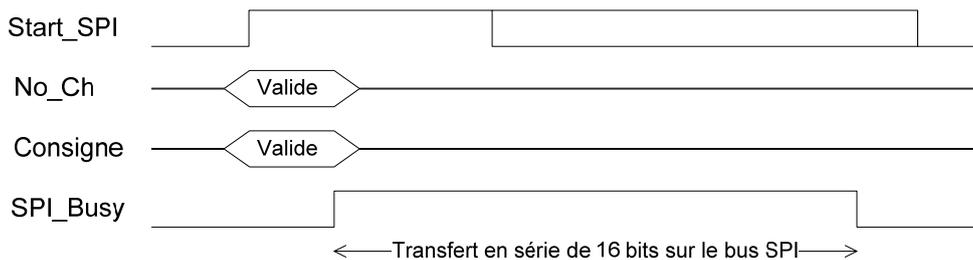
Lorsque le signal Run est inactif, la tension en sortie du convertisseur doit être à nulle (0 [V]).

Voici des spécifications sur le fonctionnement du système:

- La consigne est remise à jours toutes les 2 µs.
- La période de la tension générée aura une erreur maximum de 0.2%.
- Le fonctionnement du bus SPI est donné dans l'annexe 1.
- La fréquence de l'horloge du bus SPI est de 10 [Mhz]

Spécification de l'interface entre le générateur de consigne et le maître SPI

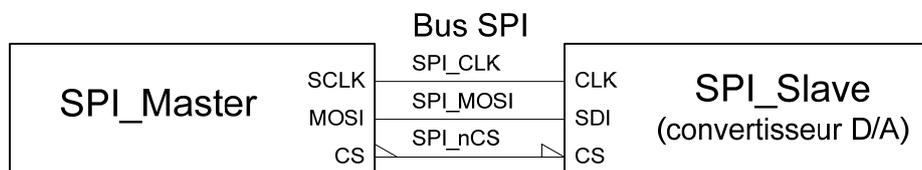
Nous vous proposons un timing pour les signaux entre les deux modules du système, soit entre le générateur de consigne et le maître SPI.



Spécification de la ligne série SPI

La spécification de la ligne série SPI est donnée dans l'annexe 1.

Dans notre cas, le slave SPI n'a pas de sortie série MISO. La figure ci-dessous montre les connexions du lien SPI.



Marche à suivre

Le projet sera réalisé en équipe par les 3 étudiants. Vous devez choisir un chef de projet qui doit répartir les tâches et établir un planning pour le projet. Les 2 autres étudiants seront responsable chacun d'un des modules du système, soit le générateur de consigne et le maître SPI. D'autre part, chacun des étudiants devra participer activement au développement du système. Vous devrez rendre un seul journal de travail pour l'ensemble du projet. Il y aura aussi deux évaluations individuelles dans le courant du travail.

Vous devez commencer par établir la liste des étapes pour la conception, le développement et la réalisation de l'ensemble du système, inclus les sous-modules. Le responsable du projet doit établir un planning du projet en concertation avec ces camarades. Ce planning doit être rendu lors de la prochaine séance. A la fin de votre travail, vous établirez une comparaison entre les durées estimées (planning) et les durées réelles. Vous devrez analyser et commenter les différences.

Vous devez prévoir dans votre démarche de valider les blocs de base que vous réalisez par une simulation manuelle avec Top_Sim et la console REDS. Il sera fourni un banc de test automatique pour le maître SPI. Vous devrez mettre en place les éléments nécessaires à une validation finale de votre solution.

Un projet textuel VHDL vous est fourni. Celui-ci comprend l'entité des composants ConvDA_SPI_top, Gen_Consigne et SPI_Master. Le projet comprend un paquetage avec la déclaration des constantes nécessaires pour le projet.

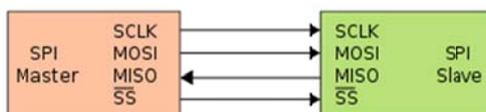
L'intégration du système sera faite dans la carte REPTAR. Vous devrez mesurer la tension obtenue en sortie du convertisseur D/A. Vous utiliserez les interrupteurs disponibles sur la carte pour le numéro du canal et les paramètres de la fonction triangle générée (amplitude et période). Un bouton poussoir sera utilisé pour le signal RUN.

Bibliographie

- [1] Spécification du bus série SPI
- [2] Data sheet convertisseur D/A d'Analog Device AD7398_7399.pdf
- [3] Reptar_reference_manual_Proto_V2.pdf
- [4] Fixed_Point_Arithmetic_.pdf

Annexe1 :**SPI: Serial Peripheral Interface****Introduction**

Une liaison SPI (pour Serial Peripheral Interface) est un bus de données série synchrone baptisé ainsi par Motorola, et qui opère en Full-duplex. Les circuits communiquent selon un schéma maître-esclaves, où le maître s'occupe totalement de la communication. Plusieurs esclaves peuvent coexister sur un bus, la sélection du destinataire se fait par une ligne dédiée entre le maître et l'esclave appelée chip select (CS ou SS).

**Interface**

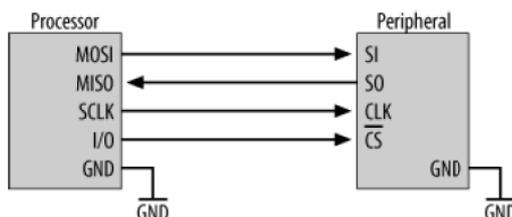
Le bus SPI contient 4 signaux logiques

- ◆ SCLK Horloge (génééré par le maître)
- ◆ MOSI Master Output, Slave Input (génééré par le maître)
- ◆ MISO Master Input, Slave Output (génééré par l'esclave), optionnel
- ◆ SS Slave Select, Actif à l'état bas, (génééré par le maître)

Il existe d'autres noms qui sont souvent utilisés.

- ◆ SCK Horloge (génééré par le maître)
- ◆ SDI, DI, SI Serial Data IN
- ◆ SDO, DO, SO Serial Data OUT
- ◆ nCS, CS, nSS, STE SS

Dans le cas de la convention de nommage SDI/SDO, le SDO du maître doit-être relié au SDI de l'esclave et vice-versa. Pour éviter les confusions au moment du câblage, il est donc souvent recommandé d'utiliser les dénominations MISO-MOSI qui évitent une certaine ambiguïté.

**Fonctionnement**

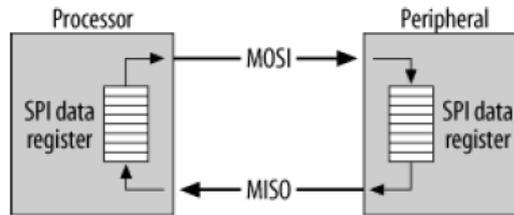
Une transmission SPI typique est une communication simultanée entre un maître et un esclave.

- ◆ Le maître génère l'horloge et sélectionne l'esclave avec qui il veut communiquer
- ◆ L'esclave répond aux requêtes du maître (optionnel)

À chaque coup d'horloge le maître et l'esclave s'échangent un bit. Après huit coups d'horloges le maître a transmis un octet à l'esclave et vice-versa. La vitesse de l'horloge est réglée selon des caractéristiques propres aux périphériques.

Le maître et l'esclave possèdent un registre à décalage. Le maître commence le transfert d'un byte en l'écrivant dans son registre à décalage. Alors que le registre transmet le byte à

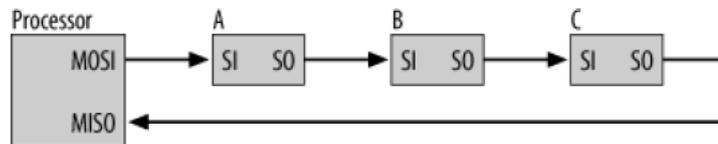
l'esclave par la ligne MOSI, l'esclave transfère le contenu de son registre à décalage en retour au maître sur la ligne MISO. De cette manière, le contenu de deux registres est échangé. Les opérations d'écriture et de lecture sont effectuées avec l'esclave en même temps. SPI peut être donc un protocole très efficace.



Si seule une opération d'écriture est désirée, le maître ignore seulement le byte qu'il reçoit. Inversement, si le maître désire seulement lire un byte de l'esclave, il doit transférer un byte factice ("dummy byte").

Quelques périphériques peuvent traiter des transferts de bytes multiples, où un flux continu de données est envoyé du maître. La plupart des mémoires avec une interface SPI fonctionnent de cette manière. Avec ce type de transfert, le "chip select" de l'esclave SPI doit rester à l'état bas pour toute la durée du transfert. Par exemple, une mémoire s'attend à recevoir une commande d'écriture suivie par quatre bytes d'adresse ("starting address"), et ensuite les données à stockées. Un simple transfert peut inclure l'envoi de grandes quantités d'informations.

D'autres esclaves ont seulement besoin d'un seul byte (par exemple, un byte de commande pour un convertisseur analogique/digital) et certains mêmes supportent être connecté ensemble ("daisy-chained").



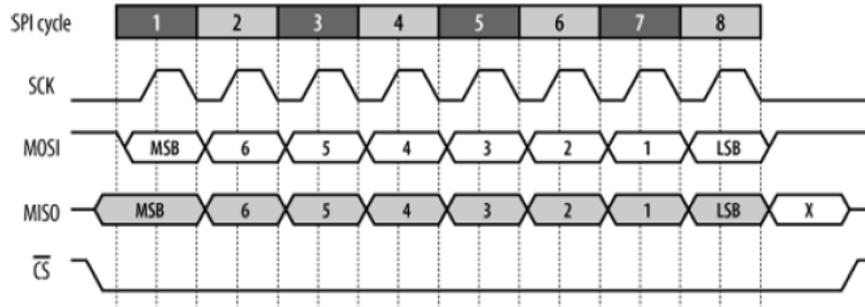
Dans l'exemple de la figure ci-dessus, le maître transmet trois bytes depuis son interface SPI. Le premier byte est reçu par l'esclave A. Comme le second byte est transféré à l'esclave A, le premier byte est transmis de l'esclave A au B. De la même manière, lorsque le troisième byte est transféré à l'esclave A, le deuxième est transféré de l'esclave A au B et le premier de l'esclave B au C. Si le maître désire lire un résultat de l'esclave A, il doit encore transférer trois bytes factices. Grâce à ça, le byte de l'esclave A sera transféré au B puis au C et enfin finalement au maître.

Il faut savoir que tous les périphériques ne fonctionnent pas forcément connectés ensemble, spécialement lorsque un nécessite des transferts multiples comme les mémoires. Le seul moyen de le savoir est de consulter la documentation du circuit.

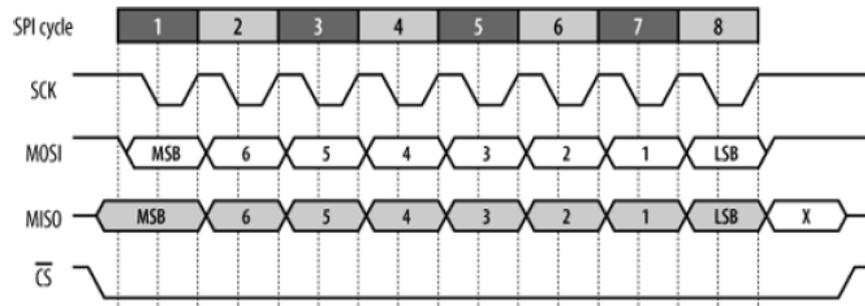
Polarité et phase de l'horloge

SPI a quatre modes opératoires. Dépendant de la polarité et de la phase de l'horloge. Pour la polarité basse de l'horloge, l'horloge SCK est à l'état bas au repos et bascule à l'état haut durant un transfert. Lorsque configuré à la polarité haute de l'horloge, l'horloge est à l'état haut au repos et bascule à l'état bas durant un transfert.

Les deux phases de l'horloge sont connues en tant que "*clock phase zero*" et "*clock phase one*". Pour *clock phase zero*, les sorties MOSI et MISO sont valides sur le front montant de l'horloge SCK si la polarité de l'horloge est basse. Si la polarité de l'horloge est haute, ces sorties sont valides sur le front descendant de SCK, pour *clock phase zero*.

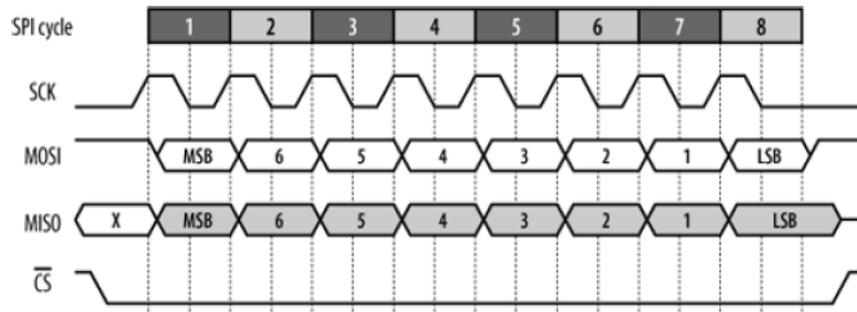


Chronogramme avec polarité de l'horloge basse et *clock phase zero*

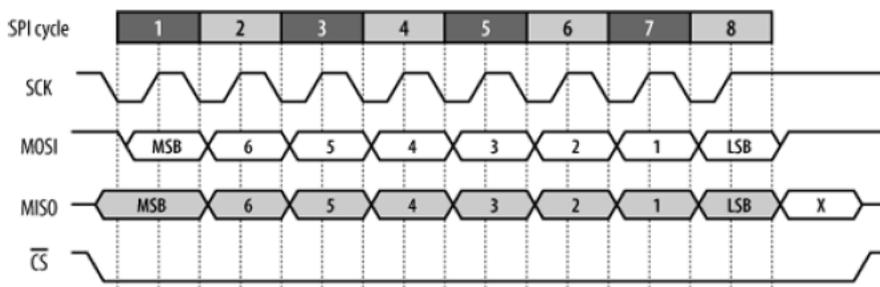


Chronogramme avec polarité de l'horloge haute et *clock phase zero*

Pour *clock phase one*, il s'agit de l'inverse de *clock phase zero*. Les données MOSI et MISO sont valides sur le front descendant de l'horloge si la polarité de l'horloge est basse. Elles sont valides sur le front montant de l'horloge si la polarité de l'horloge est haute.



Chronogramme avec polarité de l'horloge basse et *clock phase one*



Chronogramme avec polarité de l'horloge haute et *clock phase one*