HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

Prof. Yann Thoma

Laboratoire CSF
Interface SPI
semestre printemps 2016 - 2017

Objectifs pédagogiques

Ce laboratoire a pour but de réaliser une interface pour bus SPI (Serial Peripheral Interface Bus), de la vérifier et de la tester.

Cahier des charges

Le protocole SPI est un protocole série très simple à implémenter. Il permet à deux dispositifs de communiquer en s'échangeant des données au travers d'un grand registre à décalage. Il nécessite 4 fils et fonctionne avec un maître (master) et un esclave (slave).

Bus SPI

Un bus SPI est une liaison série synchrone qui opère en mode full-duplex entre un maître et un esclave. Le bus comprend trois signaux :

- **SCLK** - Serial Clock, généré par le maître
- **MOSI** - Master Out, Slave In, généré par le maître
- **MISO** - Master In, Slave Out, généré par l'esclave

Ce bus peut être utilisé pour connecter plusieurs circuits esclave. Afin de différencier le circuit avec lequel on communique un signal supplémentaire est utilisé :

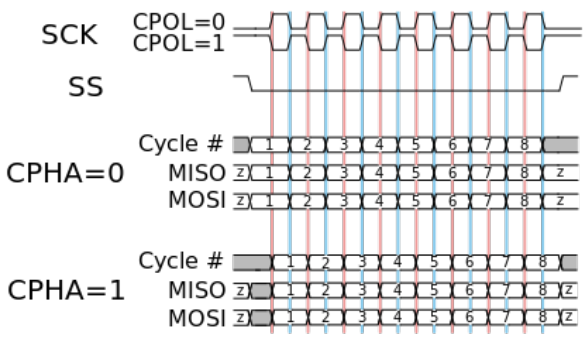
- **n_SS** - Slave Select, actif à l'état bas, généré par le maître

Le circuit maître devra générer autant de signaux slave select qu'il y a d'esclaves.

Paramètre du bus SPI

Le bus SPI peut être configuré selon les paramètres suivants :

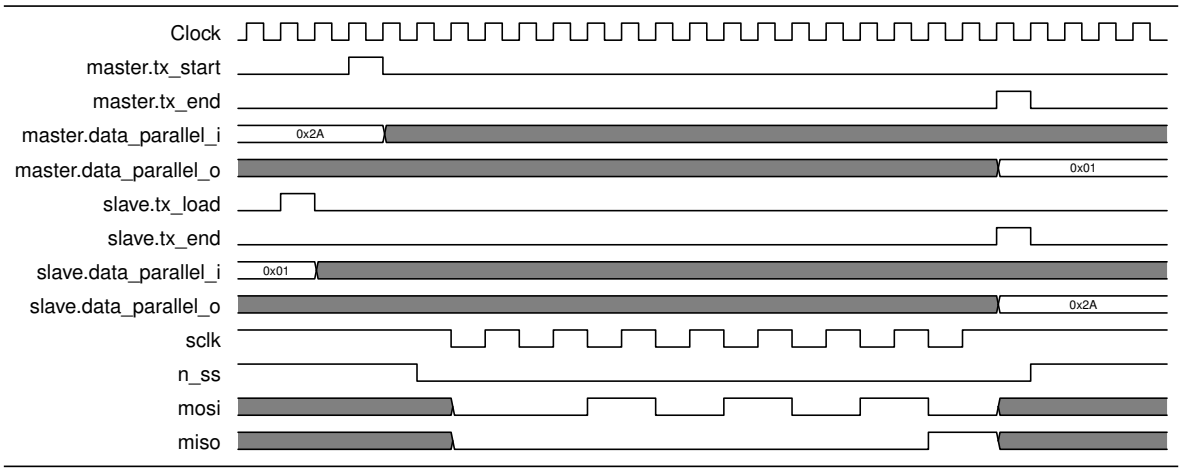
- Le nombre de bits des mots utilisés.
- La polarité de l'horloge générée par le maître.
- La phase sur laquelle les données sont écrites et lues (flanc montant ou descendant).



L'image ci-dessus montre les différentes possibilités pour une communication avec des mots de 8 bits. Pour ce laboratoire nous allons nous restreindre à utiliser la polarisation d'horloge

CPOL = 1 et CPHA = 1. Les données sont donc présentées au flanc descendant de l'horloge SCLK et lues au flanc montant.
Le nombre de bits ainsi que le diviseur d'horloge pour la génération du signal SCLK devront être ajustables grâce à des paramètres génériques. Étant donné qu'on ne réalisera qu'un maître et qu'un esclave dans le cadre de ce laboratoire on ne demandera pas de gérer la haute impédance sur le bus, en particulier pour la ligne MISO qui est partagée en écriture par les esclaves.

Exemple de communication SPI



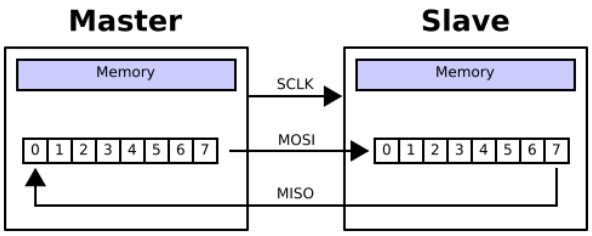
Le chronogramme ci-dessus présente une communication entre un maître et un esclave avec des mots de 8 bits. Le signal tx_start est utilisé pour initier la communication depuis le côté maître. Le maître envoie la valeur 0x2A. L'esclave ne sachant pas quand il sera accédé à déjà préparé le mot 0x01. Le maître commencer par sélectionner l'esclave grâce au signal slave select puis génère l'horloge pour la communication. Exactement 8 cycles d'horloge sont générés pour échanger les 8 bits. Une fois la communication terminée le maître désélectionne l'esclave. Finalement les deux entités lèvent un signal end afin d'indiquer au design la fin de transmission.

Circuits

Vous allez devoir réaliser deux composants, un master et un slave, qui seront ensuite exploités pour le prochain laboratoire. L'entité de chaque composant ainsi qu'un squelette de banc de test vous est fourni. A vous de réaliser le composant et de le vérifier avec le banc de test que vous modifierez en conséquence.

Implémentation

Une solution possible consiste à utiliser deux registres à décalage.



SPI Master

Entrées/sorties

Les entrées/sorties de l'entité `spi_master` sont les suivantes :

```
entity spi_master is
generic(
  N       : integer := 8;
  CLK_DIV : integer := 100 );
port (
  clk_i           : in  std_logic;
  rst_i           : in  std_logic;
  tx_start_i      : in  std_logic;
  tx_end_o        : out std_logic;
  data_parallel_i : in  std_logic_vector(N-1 downto 0);
  data_parallel_o : out std_logic_vector(N-1 downto 0);
  sclk_o          : out std_logic;
  n_ss_o          : out std_logic;
  mosi_o          : out std_logic;
  miso_i          : in  std_logic);
end spi_master;
```

- clk_i : Horloge principale pour le composant
- rst_i : Reset du composant
- tx_start_i : Permet d'initier une transmission
- tx_end_o : Indique la fin d'une transmission
- data_parallel_i : Données à transmettre
- data_parallel_o : Données reçues à la fin de la transmission

Les signaux liés au protocole SPI ont déjà été documentés. Il est à rappeler que le paramètre N permet de choisir la largeur des mots transmis et CLK_DIV permet de choisir le rapport entre l'horloge principale et l'horloge générée par le maître, si CLK_DIV = 2 l'horloge générée aura une fréquence deux fois plus basse.

SPI Slave

Les entrées/sorties de l'entité `spi_slave` sont les suivantes :

```
entity spi_slave is
generic(
  N : integer := 8 );
port (
  clk_i                    : in  std_logic;
  rst_i                    : in  std_logic;
  tx_load_i                : in  std_logic;
  tx_end_o                 : out std_logic;
  data_parallel_i          : in  std_logic_vector(N-1 downto 0);
  data_parallel_o          : out std_logic_vector(N-1 downto 0);
  sclk_i                   : in  std_logic;
  n_ss_i                   : in  std_logic;
  mosi_i                   : in  std_logic;
  miso_o                   : out std_logic);
end spi_slave;
```

- clk_i : Horloge principale pour le composant
- rst_i : Reset du composant
- tx_load_i : Permet de charger une valeur dans le registre de transmission
- tx_end_o : Indique la fin d'une transmission
- data_parallel_i : Données à charger
- data_parallel_o : Données reçues à la fin de la transmission

Les signaux liés au protocole SPI ont déjà été documentés. Il est à rappeler que le paramètre N permet de choisir la largeur des mots transmis.

Travail à réaliser

Vous devez :

1. Réaliser les deux composants
2. Réaliser un banc de test par composant pour valider leur bon fonctionnement
3. Les tester sur carte

Pour le test sur carte, un projet vous est fourni. Il permet de faire communiquer deux cartes en les connectant grâce au connecteur fourni. Le switch S9 permet de sélectionner si la carte est maître ou esclave.
Les switches S0 à S7 permettent, sur 8 bits, de saisir une valeur, autant pour le slave que pour le master. Un timer initie un transfert toutes les 200 ms depuis le maître. Un timer similaire va charger la valeur de retour dans l'esclave toutes les 200 ms. Les valeurs transférées sont affichées sur les leds 0 à 7. Si tout se passe correctement le maître affiche la valeur saisie sur l'esclave et vice-versa. Le bouton utilisé pour le reset est Key 3.

Travail à rendre

Une archive contenant le projet complet.