

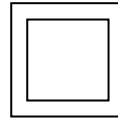
VHDL : Projet

1. Cahier des charges.

On utilise l'écran avec le protocole VGA.

A l'écran doivent apparaître 2 carrés :

Un petit et un grand (le petit doit être à l'intérieur du grand).



Lorsque l'utilisateur appuie sur le bouton poussoir 0, le grand carré doit changer de position aléatoirement (en restant dans l'écran). L'utilisateur doit alors bouger la carte DE10 pour replacer le petit carré dans le grand carré :

- Une inclinaison de la carte vers la gauche déplace le petit carré vers la gauche proportionnellement à l'inclinaison. Idem pour la droite.
- Une inclinaison de la carte vers l'avant déplace le petit carré vers le haut proportionnellement à l'inclinaison. Idem pour l'arrière et donc le bas.

Une fois le petit carré replacé dans le grand, la position du grand change de nouveau aléatoirement... Ceci durant 1 minute. A chaque fois que le petit carré a été replacé dans le grand, l'utilisateur gagne un point.

L'appui sur le bouton poussoir 1 permet de faire un reset.

2. Récupération du projet.

2.1 Le projet

Récupérez le dossier projetVhdl2024. Ouvrez le projet Quartus projetVhdl2024.qpf.

Ne pas transférer le code avant d'avoir lu les §2.1 et 2.2.

Il comporte les fichiers suivants :

Files

abc VHDL **Projet2023.vhd**

abc VHDL **PrjPack.vhd**

abc VHDL **accelSensor.vhd**

abc VHDL **accel_driver.vhd**

abc VHDL **spi_master.vhd**

abc VHDL **clock_div.vhd**

abc VHDL **decoder7seg.vhd**

- *Projet2024.vhd* est le top level entity.
- *PrjPack.vhd* est le pack qui comporte pour l'instant 2 composants (**ne modifiez pas ces composants**). Vous pourrez ajouter autant de composants que vous voulez.
- *accelSensor.vhd*, *accel_driver.vhd*, *spi_master.vhd* et *clock_div.vhd* sont les fichiers qui gèrent l'accéléromètre. En aucun cas vous ne devez modifier ces fichiers.
- *decoder7seg.vhd* est le fichier qui gèrent un afficheur 7 segments. Il est donné (cadeau de Noël), **ne le modifiez pas**, il fonctionne très bien.

VOUS NE DEVEZ PAS MODIFIER LE PIN PLANNER. TOUS LES SIGNAUX À UTILISER SONT DÉJÀ DÉFINIS ET MAPPÉS.

2.2 decoder7seg.vhd

Voici son entité :

```
stv4_int_data : in std_logic_vector(3 downto 0);
st_int_en : in std_logic;
stv7_out_seg : out std_logic_vector(6 downto 0)
```

Le nombre N codé en binaire sur stv4_int_data est transcodé dans stv7_out_seg afin d'allumer les bons segments d'un afficheur 7 segments pour afficher le nombre N (de 0 à 9 puis A, b, C, d, E, F pour respectivement 10 11 12 13 14 15). L'entrée de validation du composant (st_int_en) est active à l'état bas.

decoder7seg est déjà présent dans le pack.

2.3 accelSensor.vhd

accelSensor.vhd instancie accel_driver.vhd, spi_master.vhd et clock_div.vhd afin de récupérer les inclinaisons de la carte DE10 en X et Y (via la capteur accéléromètre de la carte). L'entité de accelSensor est :

```
st_in_clk50MHz : in std_logic;
st_out_mosi : out std_logic;
st_in_miso : in std_logic;
st_out_sclk : out std_logic;
st_out_cs : out std_logic;
st_in_int1 : in std_logic;
st_in_intBypass : in std_logic;
```

Ne pas toucher : Ces signaux gèrent la liaison I2C entre le FPGA et le capteur *accéléromètre*. Une fois accelSensor instancié correctement, le capteur fonctionne en permanence. Les données stv4_out_dataX, stv4_out_sensX, stv4_out_dataY et stv4_out_sensY sont donc disponibles en permanence et en temps réel. Tout est déjà fait (re-cadeau de Noël). Ces signaux sont mappés dans le pin planner. Ne pas modifier ce mapping.

```
stv4_out_dataX : out std_logic_vector(3 downto 0); -- Donne l'inclinaison de la carte dans le sens X
stv4_out_sensX : out std_logic_vector(3 downto 0); -- Donne le sens de l'inclinaison de la carte dans le sens X
stv4_out_dataY : out std_logic_vector(3 downto 0); -- Donne l'inclinaison de la carte dans le sens Y
stv4_out_sensY : out std_logic_vector(3 downto 0); -- Donne le sens de l'inclinaison de la carte dans le sens Y
```

accelSensor est déjà présent dans le pack et déjà instancié dans Projet2024.vhd et mappé dans le pin planner.

2.4 Projet2024.vhd

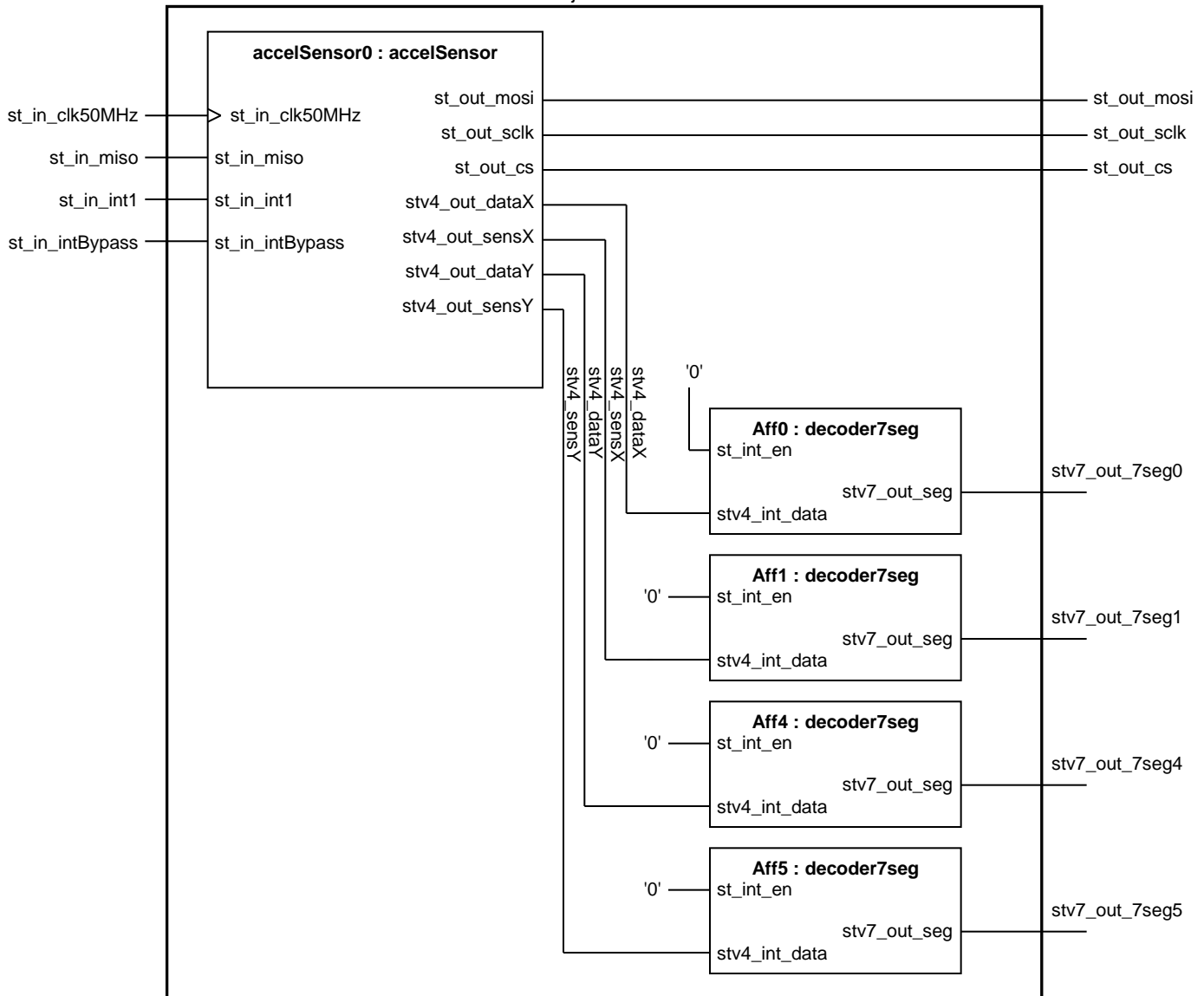
C'est le fichier dans lequel vous allez instancier vos composants pour votre projet ou en cours de projet pour les tester.

Vous ne devez pas modifier l'entité qui contient tous les signaux utilisables de la carte. Vous utiliserez ceux que vous voudrez. Vous laisserez ceux que vous n'utilisez pas, cela ne génère aucune erreur... Ces signaux sont déjà mappés au bon endroit dans le pin planner. **Vous ne devez en aucun cas modifier le pin planner (je pense que vous avez maintenant compris...).**

Vous pouvez réutiliser le signal d'horloge 50MHz lors d'une instanciation d'un de vos composants.

Afin de tester le bon fonctionnement de la carte et des descriptions vhdl déjà existantes, nous (enfin je) avons réalisé cette première instanciation :

Projet2024



Téléversez le projet dans la carte DE10. Testez. Notez quel axe est l'axe X, quel axe est l'axe Y, quels sont les inclinaisons min et max et comment est indiqué le sens de l'inclinaison. Ça peut être utile pour le projet...

Pour le projet il est conseillé de conserver l'affichage des données inclinaison X et Y sur 2 afficheurs 7 segments. L'affichage des sens d'inclinaison pourront être codés sur une LED afin de récupérer 2 afficheurs pour les points gagnés et 2 afficheurs pour le décompte de la minute.

Dans le fichier Projet2024.vhd vous pouvez faire toutes les instanciations que vous voulez et ajouter autant de signaux internes que vous voulez.

2. Méthode.

Vous pouvez placer dans le dossier projetVhdl2024 les fichiers VHD qui gèrent les signaux VGA des TP précédents (CompteurX, CompteurY, GeneRGB...). Ajoutez-les à votre projet. Il est conseillé de les instancier dans Projet2024.vhd et de tester leur bon fonctionnement dans le projet en cours (carré qui bouge par exemple).

Pour l'aléatoire, il est conseillé de fabriquer un compteur sur horloge 50MHz. Comme on ne sait pas à quel moment l'utilisateur va appuyer sur le bouton poussoir et en combien de temps il va placer le petit carré dans grand, la donnée en sortie du compteur est forcément aléatoire (on l'appelle pseudo aléatoire dans ce cas).

Vous pouvez aussi tester le côté aléatoire et les boutons poussoirs en utilisant les afficheurs... Pour cela instanciez toujours les composants à tester dans l'instance supérieure : Projet2024.vhd

Il est vivement conseillé de ne pas tout coder d'un coup. Faites le schéma des instanciations et des liaisons entre les composants instanciés. Cela vous donnera la liste des composants à créer. Créez ensuite chaque composant et n'hésitez pas à les tester les uns à part des autres : Utilisez les afficheurs et les LED pour tester et éventuellement les sorties Arduino avec la sonde et l'oscilloscope.

Pour tester instanciez toujours les composants à tester dans l'instance supérieure : Projet2024.vhd

Vous pouvez aussi utiliser un graphe d'état (c'est un peu plus qu'un conseil en fait). La méthode pour coder un graphe d'état est donné sur U:\Documents\BUT\GEII\ModulesS5\VHDL\VhdlTp2.pdf.

Bon courage...

Pour vos questions : thierry.suaton@univ-smb.fr

Rendu : 31 janvier 2025.

Un compte rendu est demandé qui comportera uniquement le schéma des instanciations et des liaisons entre les composants instanciés.

Des améliorations pourront être ajoutées pour personnaliser votre projet.

Joyeux Noël et Bonne année !!!