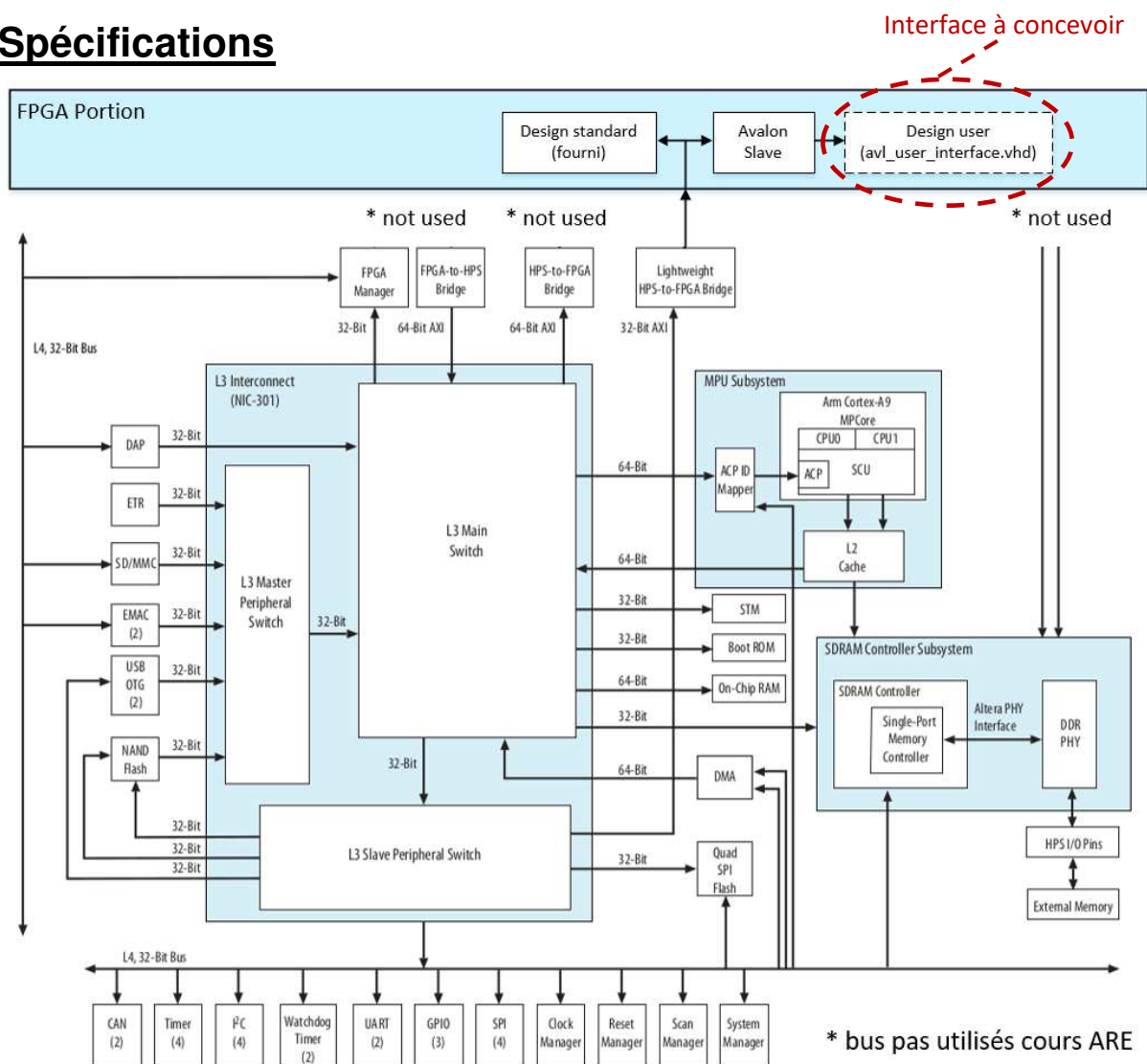


Laboratoire 03 – Conception d'une interface simple

Objectifs du laboratoire

Ce laboratoire a pour but de concevoir une interface simple sur le bus Avalon connecté sur le système à processeur HPS (hard processor system). Vous devrez définir votre plan d'adressage pour accéder et commander les différents périphériques. Les périphériques que vous utiliserez seront des E/S de la carte DE1-SoC (Leds, boutons, interrupteurs) et une liaison parallèle avec 36 liens direct de la carte Max10_leds. Ensuite, vous écrirez un programme pour piloter votre interface afin de respecter les spécifications demandées.

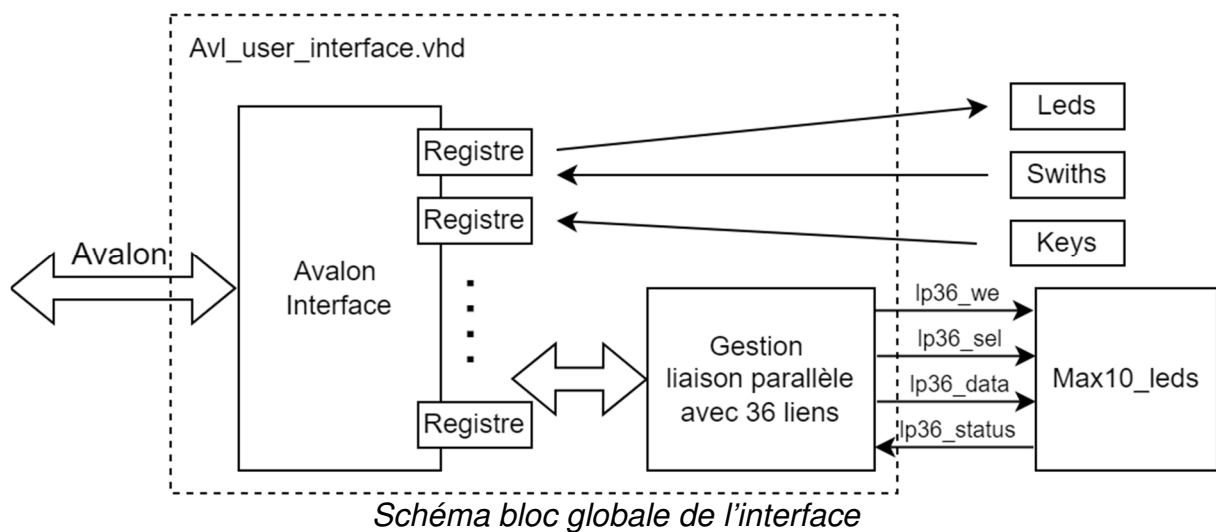
Spécifications



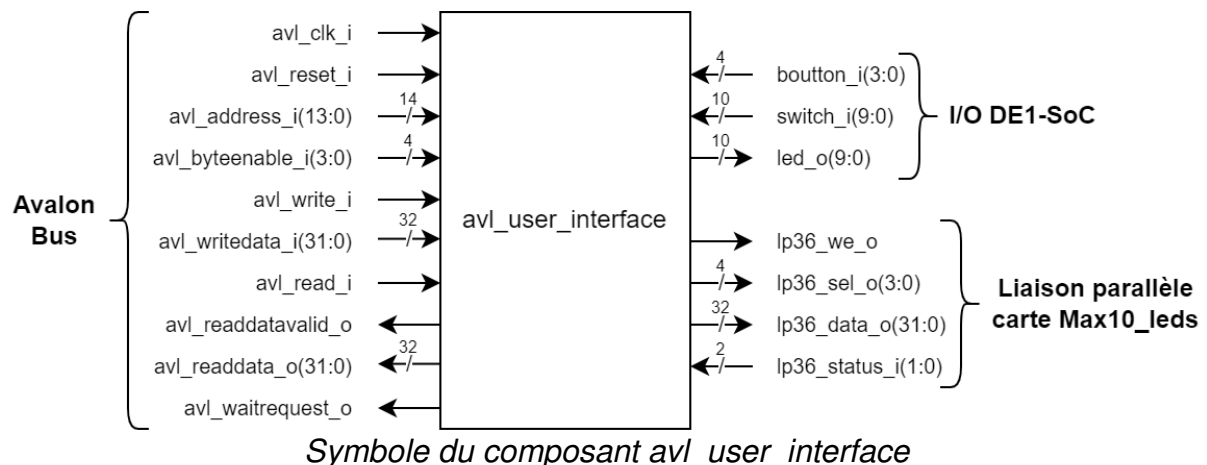
Représentation du HPS avec le composant `avl_user_interface`

Le composant VHDL "`avl_user_interface.vhd`" sera utilisée pour implémenter l'interface connecté sur le bus Avalon afin de commander les différents périphériques.

Voici le schéma bloc globale de l'interface à réaliser :



Voici le symbole du composant avl_user_interface :



Pour comprendre le fonctionnement du bus Avalon, une description est fournie dans le document "Description_Bus_Avalon.pdf". L'horloge du bus Avalon est à 50Mhz.

L'interface développée doit permettre d'accéder, à travers le bus Avalon, aux périphériques suivants :

- Interface user ID sur 32 bits (disponible à l'offset 0)
- 4 Boutons (Key) DE1-SoC
- 10 Interrupteurs (Switch) DE1-SoC
- 10 Leds DE1-SoC
- Liaison parallèle avec 36 liens direct de la carte Max10_leds (sur le connecteur 80 pôles). Détails donnés ci-après.

Plan d'adressage

Voici le plan d'adressage qui est spécifié pour le bus AXI lightweight HPS-to-FPGA du projet fourni.

| Offset on bus AXI lightweight HPS-to-FPGA (relative to BA_LW_AXI) | Fonctionnalités |
|--|--|
| 0x00_0000 – 0x00_0003 | Design standard ID 32 bits (Read only) |
| 0x00_0004 - 0x00_FFFF | reserved |
| 0x01_0000 - 0x01_FFFF | Zone disponible pour votre interface |
| 0x02_0000 - 0x1F_FFFF | not used |

Liaison parallèle avec 36 liens direct de la carte Max10 leds

Cette liaison parallèle permet de contrôler l'état des leds sur la carte Max10_leds. Un signal de statut sur 2 bits permet d'informer que la carte Max10_leds est dans le bon mode de fonctionnement. Un signal de sélection sur 4 bits permet de sélectionner la zone de leds à contrôler, et une donnée sur 32 bits vient déterminer l'état allumé ou éteint des dites leds. Un signal de donnée valide (write_enable) permet l'écriture sur ce bus parallèle. Pour avoir une écriture fiable, ce signal de donnée valide doit être actif pendant au moins **1us** et cela représente un cycle d'écriture. Les tableaux ci-dessous décrivent d'une part les signaux de cette liaison parallèle, et d'autre part le codage permettant d'accéder aux leds présentes sur la carte.

| Noms des signaux | Source | Description |
|-------------------------|---------------|---|
| lp36_status (1..0) | Max10 | Signal de statut de la carte Max10_leds, sur 2 bits : <ul style="list-style-type: none"> - 00 : configuration non valide. - 01 : configuration valide. - 1X : réservé |
| lp36_sel (3..0) | Interface | Signal de sélection des leds pilotées, sur 4 bits. Voir le tableau ci-dessous pour la définition du codage. |
| lp36_data (31..0) | Interface | Signal de commande des leds, sur 32 bits, selon la valeur de la sélection. |
| lp36_we | Interface | Signal de write enable pour une écriture sur le bus. |

Tableau du codage du bus de data (lp36_data) selon la valeur du groupe sélectionné (lp36_sel) :

| lp36_sel (3..0) | Spécification de lp36_data (31..0) |
|-----------------|---|
| 0000 | lp36_data31-30 non utilisés, Leds secondes DS30...1 |
| 0001 | lp36_data31-30 non utilisés, Leds secondes DS60...31 |
| 0010 | Les 2 lignes de leds DL31...16 & DL15....0 |
| 0011 | lp36_data 31-25 non utilisés, Carré de leds DM11-15/21-25/31-35/41-45/51-55 |
| 0100 | réservé |
| ... | ... |
| 1111 | réservé |

Spécifications du programme

Le but est de contrôler les différentes leds de la carte Max10_leds selon l'états des boutons et interrupteurs de la DE1-SoC. La spécification du fonctionnement est la suivante :

Au démarrage, le programme doit remplir les conditions suivantes :

- Vérifier que le statut de la carte Max10_leds est une configuration valide. Sinon afficher un message d'erreur dans la console ARM-DS et quitter le programme.
- Les 10 leds DE1-SoC sont éteintes.
- Toutes les leds de la carte Max10_leds sont éteintes (leds secondes, 2 lignes de leds, carré de leds).
- Afficher la constante ID du bus AXI lightweight HPS-to-FPGA au format hexadécimal dans la console de ARM-DS.
- Afficher la constante ID de votre interface sur le bus Avalon au format hexadécimal dans la console de ARM-DS.

Ensuite pendant l'exécution du programme, à tout instant les actions suivantes doivent être respectées :

- Copie de la valeur des 10 interrupteurs (SW) sur les 10 leds de la DE1-SoC.
- L'état de SW9-8 permet de sélectionner les leds à mettre à jour sur la carte Max10_leds :
 - o SW9-8 = 00 : Leds secondes DS30.. .1.
 - o SW9-8 = 01 : Leds secondes DS60...31.
 - o SW9-8 = 10 : Les 2 lignes de leds DL.
 - o SW9-8 = 11 : Le carré des leds DM.
- L'état de KEY1-0 permet de définir la valeur affichée sur les leds sélectionnés de la carte Max10_leds :
 - o KEY1-0 = 00 : Copie de la valeur des 8 interrupteurs (SW0 to SW7) sur les poids faibles. Les leds de poids forts sont éteintes.
 - o KEY1-0 = 01 : Afficher la valeur 1010...1010.
 - o KEY1-0 = 10 : Afficher la valeur 0101...0101.
 - o KEY1-0 = 11 : Afficher la valeur 1111...1111.
- Pression sur KEY2 :
 - o Lors de la sélection du carré des leds DM ainsi que la copie de la valeur des 8 interrupteurs : Faire décaler d'une ligne vers le bas la valeur des 8 interrupteurs affichés sur le carré des leds DM.
- Pression sur KEY3 :
 - o Eteindre toutes les leds de la carte Max10_leds.

Travail demandé

- 1) Etudier le fonctionnement du bus Avalon à l'aide du document fourni "Description_Bus_Avalon.pdf". Dans le plan d'adressage, la taille de la zone disponible pour votre interface correspond telle aux 14 bits d'adresse défini dans le bus Avalon ? Pourquoi ?
- 2) Définir un plan d'adressage pour votre interface afin de répondre au besoin de la spécification. Expliquez votre choix.
Cette définition doit permettre d'avoir un accès facile aux différents périphériques. Vous disposez d'une zone mémoire très importante.
- 3) Vous devez faire valider votre plan d'adressage auprès du professeur ou de l'assistant.
- 4) Concevoir le schéma bloc de l'interface Avalon permettant de répondre aux différents accès prévus dans votre plan d'adressage (Composant "Avalon interface" du schéma bloc globale en page 2). Etudier comment vous pouvez décrire ce schéma bloc à l'aide d'une description VHDL et éventuellement adapter votre schéma bloc.
- 5) Concevoir le schéma bloc qui représente la gestion de la liaison parallèle avec la carte Max10 (Composant "Gestion liaison parallèle avec 36 liens" du schéma bloc globale en page 2). Faire apparaître les registres utiles pour la liaison parallèle.
- 6) Décrire en VHDL synthétisable votre interface, conçu dans les points précédents, dans le fichier "avl_user_interface.vhd".
- 7) Simuler votre interface à l'aide de la console TCL/TK (voir document Simulation_console_TCL-TK.pdf pour son utilisation). Vous ajouterez dans votre rapport une capture d'écran de la simulation (chronogramme), la listes des commandes utilisées et une explication des résultats de la simulation.
- 8) Tester votre interface sur la carte DE1-SoC. Vous utiliserez l'accès mémoire pour vérifier son bon fonctionnement. Expliquer vos tests et résultats dans le rapport.

Nous vous conseillons d'avancer progressivement dans le développement de votre interface (points 6, 7 et 8). Faites des vérifications et des tests étapes par étapes.

- 9) Ecrire des fonctions C pour les accès aux différents périphériques. Puis écrire le programme principal suivant les spécifications.
- 10) Tester vos fonctions et le programme sur la carte DE1-SoC. Expliquer vos tests et résultats dans le rapport.
- 11) Faire valider votre programme par le professeur ou l'assistant.

À rendre

Ce laboratoire est évalué. Il y a un rapport à rédiger à l'issu de ce laboratoire contenant les explications de toutes les étapes de la réalisation de votre système. Vous devez rendre une archive avec les sources du projet pour Quartus et le programme C. Utiliser le Makefile à la racine du projet pour générer votre archive à rendre en tapant « make zip » dans un terminal.

Les fichiers sont à rendre sur Cyberlearn à la date indiquée.