Tutoriel pour l'utilisation de l'afficheur LCD du Genesys dans l'environnement Active-HDL

Philippe Proulx

Août 2011
École Polytechnique de Montréal

Introduction

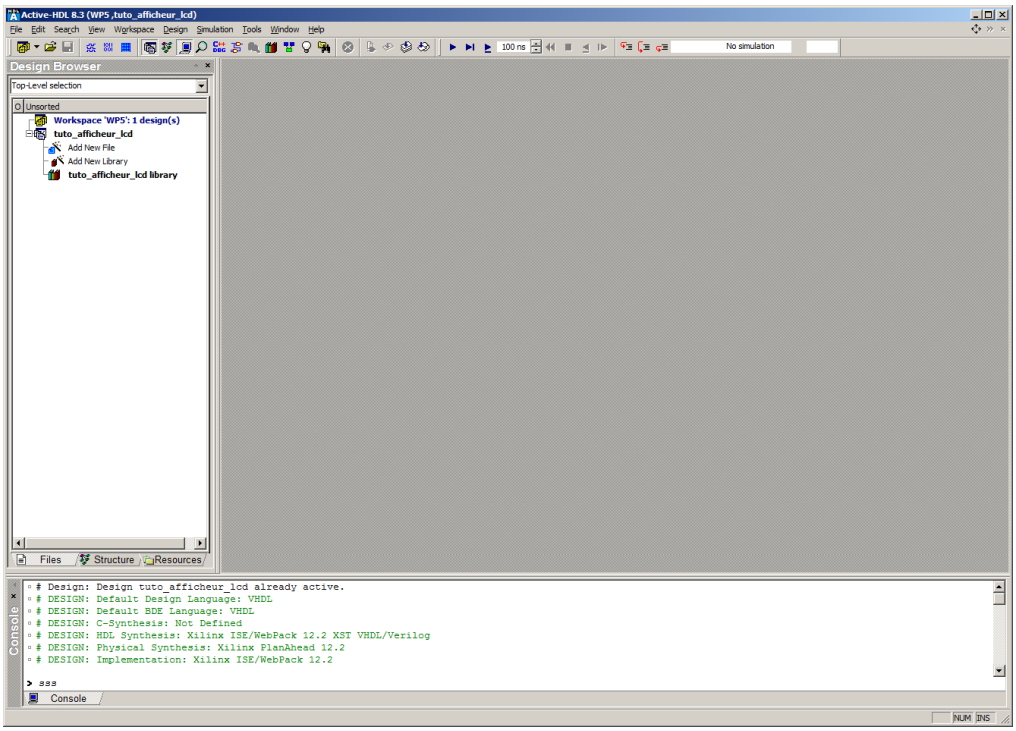
L'afficheur LCD du Digilent Genesys, la carte de développement installée dans les laboratoires L-3712 et L-3714 de l'École Polytechnique de Montréal, est assez complexe à utiliser directement pour un débutant en logique numérique. Ainsi, des modules d'aide ont été développés afin de profiter d'une interface simple d'utilisation pour ce périphérique.

Ce document vous guide à travers les étapes pour créer un design schématique permettant l'affichage d'une valeur sur 4-bit en hexadécimal dans une case de l'afficheur LCD. Cette valeur sera déterminée par les 4 premiers interrupteurs (les plus à droite sur le Genesys).

Vous devriez avoir suivi et compris le guide « INF1500 – logique des systèmes numériques : introduction à l'utilisation d'Active-HDL 8.3sp1 et à la technologie FPGA » avant de vous lancer dans la lecture de ce document.

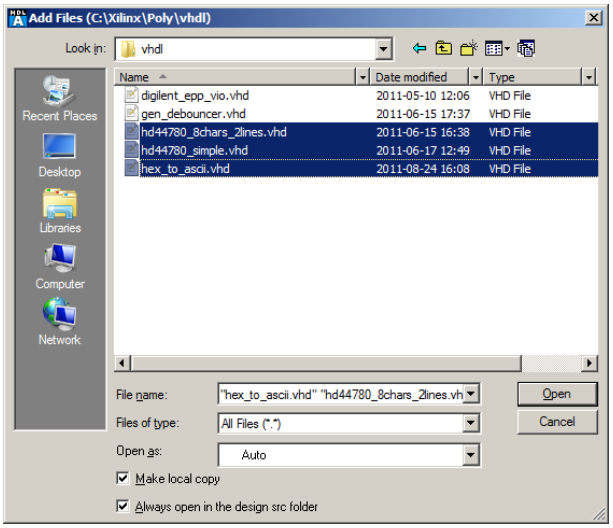
Tutoriel

Nous supposons ici que votre *workspace* est déjà créé et que vous y avez ajouté un design. Vous avez donc cette vue, mise à part les noms de *workspace* et de design :

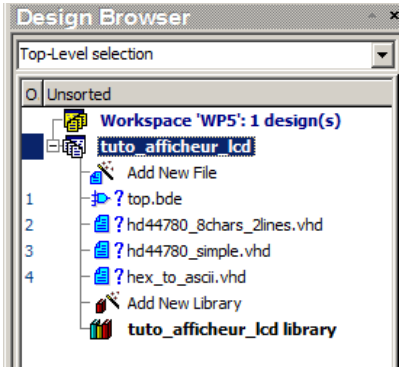


Avant toute chose, ajoutez un nouveau bloc schématique nommé `top`. Le fichier créé se nommera alors `top.bde`. Ensuite, cliquez droit sur le nom du design (dans notre cas `tuto_afficheur_lcd`) et sélectionnez « Add Files to Design… ». Allez ensuite chercher les trois fichiers nécessaires `hd44780_8chars_2lines.vhd`, `hd44780_simple.vhd` et `hex_to_ascii.vhd`

situés dans le répertoire `C:\Xilinx\Poly\vhdl` (maintenez la touche du clavier CTRL enfoncée pour sélectionner plusieurs fichiers) :

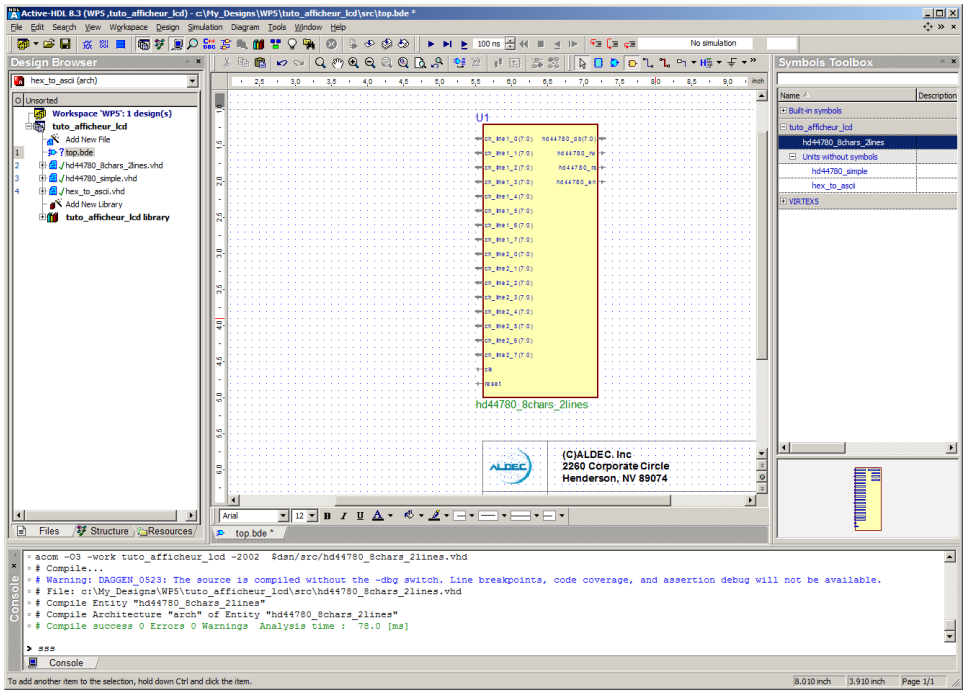


Cochez également l'option « Make local copy » afin que les fichiers sélectionnés soient copiés dans le répertoire de votre design. Les trois fichiers seront alors inclus dans votre design :



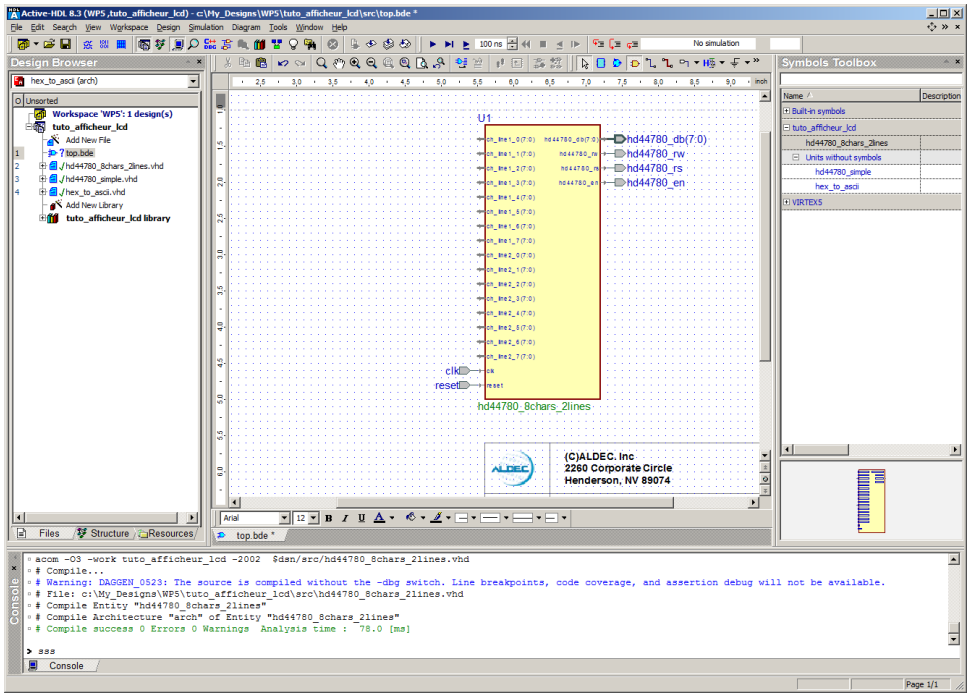
Cliquez droit sur `hex_to_ascii.vhd` et sélectionnez « Compile ». Un petit crochet vert apparaitra à gauche de son nom. Faites de même pour les deux autres fichiers ajoutés.

Double-cliquez maintenant sur `top.bde` afin d'ouvrir l'éditeur schématique. Appuyez sur la touche « S » pour afficher la boite de symboles et glissez le symbole `hd44780_8chars_2lines` (sous `tuto_afficheur_lcd`) sur votre schéma :



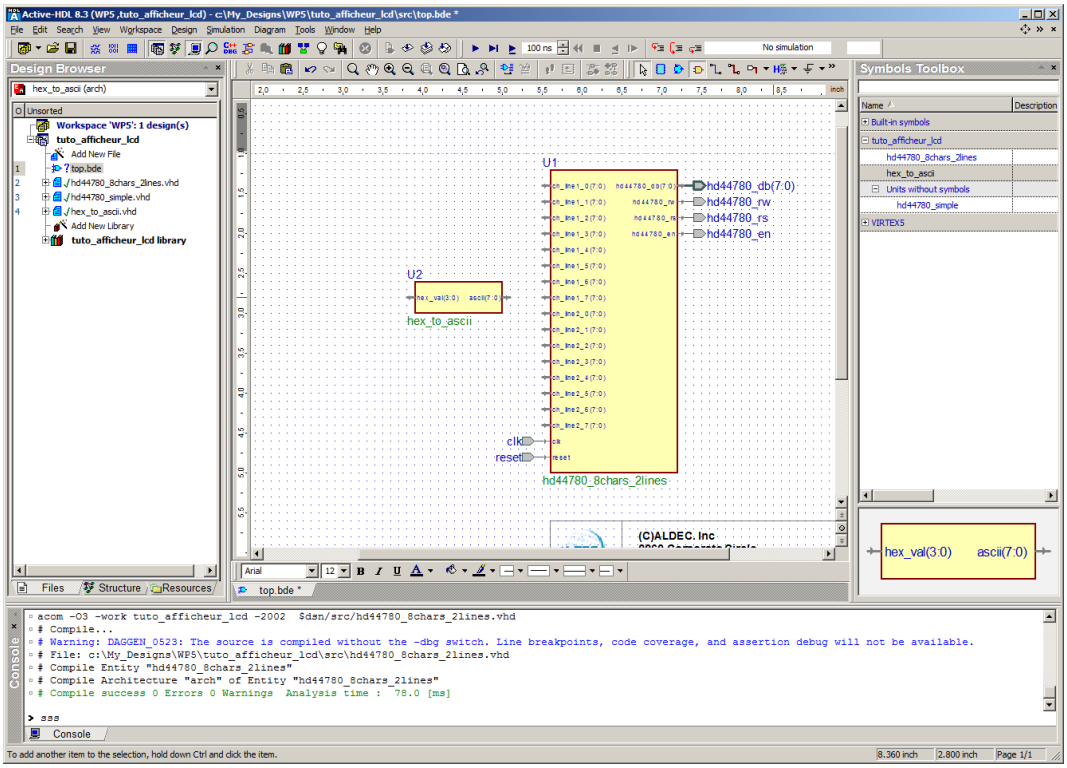
Ce module prend en entrée une horloge, un signal de réinitialisation et 16 bus de 8 bits correspondant à 16 caractères compatibles ASCII sur l'afficheur LCD. Les 8 premiers bus représentent les 8 caractères centrés de la première ligne et les 8 autres ceux de la ligne suivante. En sortie, on retrouve des signaux propres à l'afficheur LCD qui devront être localisés vers des broches spécifiques à l'aide du fichier UCF.

Ajoutez toutes les sorties (simples et de bus) et les entrées pour `clk` et `reset`. Les noms des entrées/sorties devraient automatiquement être les mêmes que ceux des ports où ils sont connectés si vous les connectez directement. Sinon, renommez-les pour qu'ils le soient :

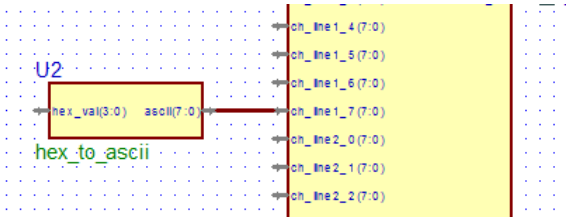


Bien que le fait de pouvoir afficher n'importe quel caractère ASCII soit intéressant, en INF1500, nous sommes plus souvent qu'autrement intéressés par l'affichage d'une valeur. Or, si vous ajoutez un bus à l'une des entrées de caractère (8-bit) et que vous y mettez, par exemple, la valeur 3, vous ne verrez rien d'intéressant sur l'afficheur puisque le caractère ASCII 3 est un caractère de contrôle et ne correspond pas au caractère « 3 ». Le caractère « 3 » possède plutôt le code ASCII 51.

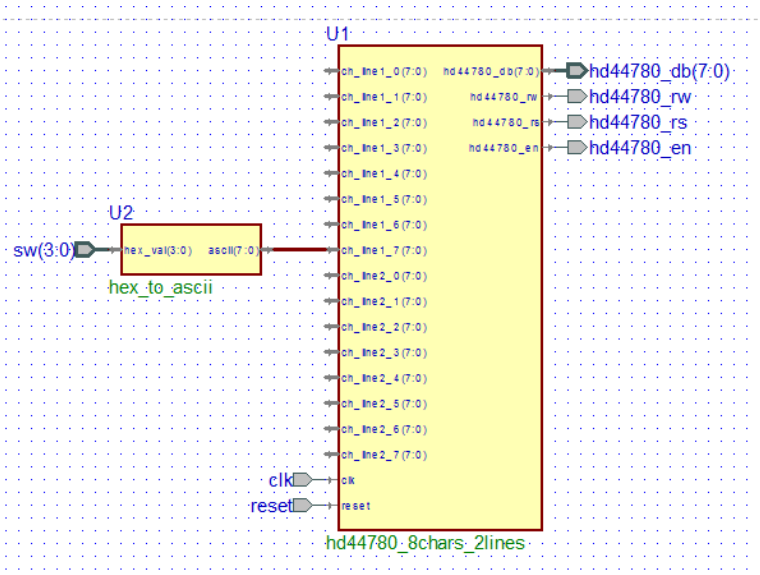
Afin d'éviter aux étudiants le fardeau de convertir d'une valeur à un caractère ASCII, le module `hex_to_ascii` est fourni. Ce module prend en entrée une valeur hexadécimale sur 4 bits et produit en sortie le caractère ASCII correspondant à cette valeur (« 0 » à « 9 » et « a » à « f »). Glissez donc ce symbole, à partir de la boite de symboles à droite, à gauche du module de l'afficheur LCD :



Ajoutez ensuite un bus entre la sortie de `hex_to_ascii` et l'entrée d'un des caractères ASCII du module de l'afficheur LCD (à votre guise) :



Enfin, ajoutez une entrée de bus à l'entrée inoccupée du module `hex_to_ascii`. La largeur de ce bus devrait être de 4 bits et sera automatiquement configurée ainsi si vous connectez l'entrée directement au module. Renommez ce bus `sw` :



Le design est maintenant complet. Appuyez sur « F11 » pour le compiler. Vous êtes prêt à le synthétiser, à l'implémenter et à l'envoyer vers la carte de développement comme d'habitude. En guise de référence, utilisez ces entrées pour le fichier UCF :

```
net clk loc = ag18;
net clk tnm_net = clk;
timespec ts_clk = period "clk" 100 MHz HIGH 50%;

net reset loc = g6;
net hd44780_en loc = aa5;
net hd44780_rs loc = v7;
net hd44780_rw loc = w6;
net hd44780_db<0> loc = y8;
net hd44780_db<1> loc = ab7;
net hd44780_db<2> loc = ab5;
net hd44780_db<3> loc = ac4;
net hd44780_db<4> loc = ab6;
net hd44780_db<5> loc = ac5;
net hd44780_db<6> loc = ac7;
net hd44780_db<7> loc = ad7;
net sw<3> loc = h18;
net sw<2> loc = k18;
net sw<1> loc = l18;
net sw<0> loc = j19;
```

En changeant la valeur des interrupteurs, vous voyez la valeur changer sur l'afficheur LCD. Le tour est joué!

Comme vous pouvez le constater, vous n'êtes pas obligé de connecter toutes les entrées du module de l'afficheur LCD. Les caractères non connectés prendront alors une valeur indéterminée (parfois un caractère entièrement noir).