

TP1 : Compteur BCD affichage 7 segments

Objectifs

- Implémenter un compteur BCD (logique synchrone) et un décodeur 7-segments en VHDL (logique combinatoire).
- Vérifier le compteur par simulation logique (ModelSim).
- Synthétiser le projet complet.
- Télécharger le projet sur la carte

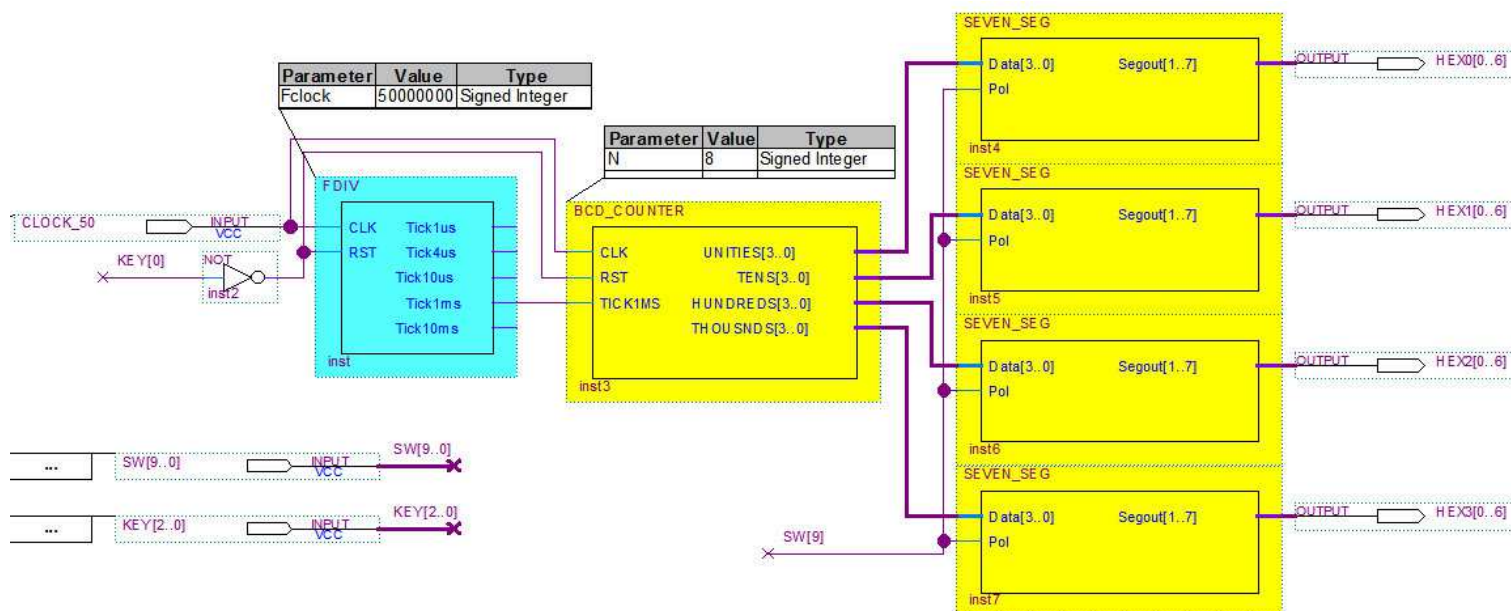
Cet exercice doit afficher un compteur de 0 à 9999 sur les afficheurs 7-segments de la carte. Seul le compteur et le décodeur sont à écrire, le niveau supérieur de cette application étant fourni. L'affichage et la génération des tops à 1 ms est pris en charge, et un banc de test unitaire est également fourni.

Malgré sa simplicité, cet exercice permet de mettre en oeuvre l'essentiel de la méthodologie et des outils de conception.

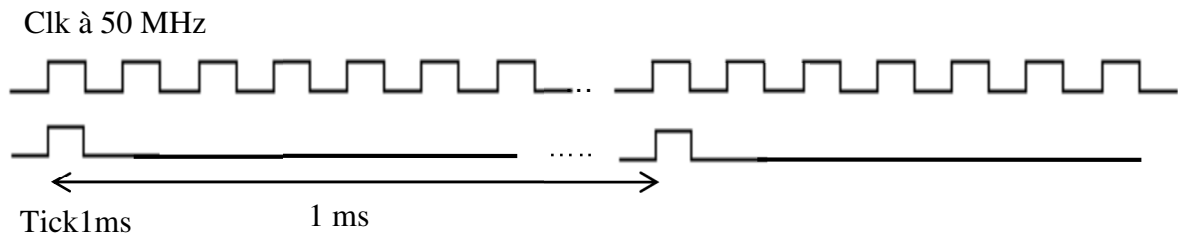
Description de l'exercice

L'horloge d'entrée du FPGA est une horloge de 50 MHz. On désire compter de 0000 à 9999 au rythme d'une incrémentation toutes les « **N** » **millisecondes**, N étant paramétrable à l'instanciation par un paramètre « **generic** » : **N**. Le compteur doit repasser à 0000 lorsque l'on se trouve à 9999.

Voici le schéma descriptif de la fonctionnalité demandée :



Le bloc FDIV fournit un signal **Tick1ms** actif (niveau logique '1') toutes les 1 millisecondes durant un cycle d'horloge à 50 MHz.



Le signal **Tick1ms** arrive sur un port d'entrée du bloc BCD_Counter et doit être utilisé pour réaliser l'incrémentación du compteur toutes les **N ms**.

Il est hors de question d'utiliser Tick1ms comme horloge (l'horloge est toujours Clk, l'horloge principale !!!) Il vous faudra réaliser un test sur l'état de Tick1ms.

Le compteur BCD a 4 ports de sortie : un pour les unités, un pour les dizaines, un pour les centaines et le dernier pour les milliers. Chacun de ces ports est bien sûr codé sur 4 bits, afin de pouvoir compter de 0 à 9 inclus ... (et pas au-delà).

Ces ports sont ensuite encodés par le bloc SEVEN_SEG pour pouvoir être affichés sur les afficheurs 7 segments.

Décodeur 7-segments

Voici le fonctionnement d'un afficheur 7-segments :

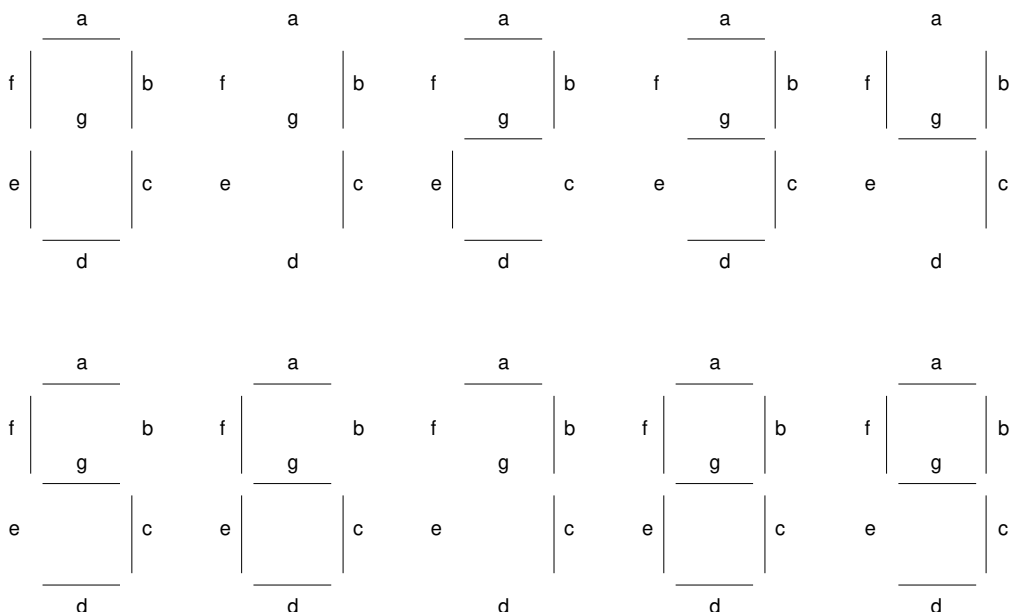
Les segments sont traditionnellement identifiés par les lettres de A à G (ou « a » .. « g ») suivant la convention représentée. Dans notre module, la sortie est un vecteur croissant de 1 à 7 correspondant aux segments A à G.

De plus, nous avons une entrée de sélection de polarité « Pol » :

Pol = 1 signifie des afficheurs actifs niveau 1

Pol = 0 signifie des afficheurs actifs niveau 0

7-Segments Decoder - BCD (0..9) only



Ainsi, pour afficher le chiffre 1, on devra établir SegOut = 0110000 si Pol=1 et 1001111 si Pol=0.

Ce qu'il faut faire

Copier le dossier TP1_BCD.zip

Ce répertoire est sectionné en 3 sous répertoires qui séparent bien les trois étapes nécessaires pour la réalisation complète d'un design (édition, simulation et synthèse) :

- **src** : qui contient les fichiers sources VHDL nécessaires pour le design (dont les fichiers à compléter).
- **simu** : qui contient les fichiers nécessaires à la simulation (Test Bench, script de simulation, modèle VHDL, etc....)
- **fit** : qui contient les fichiers nécessaires à la synthèse (Script de synthèse, rapport de synthèse, placement / routage, analyse des Timing, etc...)

Pour l'ensemble du TP1, vous allez utiliser **NotePad ++** comme éditeur de VHDL et **Modelsim** pour la simulation. Un ensemble de scripts sont fournis et permettent de ne pas perdre trop de temps pour la simulation.

Écrire le compteur BCD en VHDL

- > Ouvrez le fichier BCD_COUNTER.vhd sous le répertoire TP1_BCD/src. Ce fichier contient déjà la déclaration des ports de l'entité BCD_COUNTER, et également la déclaration de l'architecture de l'entité ainsi que le modèle de process synchrone avec reset asynchrone. Vous devez insérer votre code là où les commentaires l'indiquent, et pas ailleurs. Ceci permet d'obtenir un code qui respecte les règles usuelles de codage, de synthèse, et de qualité.
- > Codez un compteur BCD qui compte toutes les N x 1 ms de 0000 à 9999. N'oubliez pas que le compteur doit repasser à 0000 lorsque l'on est à la valeur 9999...

Vérifier le compteur (par simulation RTL)

- > Lancer ModelSim.
- > Dans Modelsim, aller sous le répertoire TP1_BCD/simu (File > Change Directory ou commande **cd**).
- > Taper la commande : "**do simu.do**" qui lance le script de compilation et de simulation.
- > Si il y a une erreur de compilation, analyser la nature de l'erreur, corriger (attention à ne pas ouvrir le fichier avec plusieurs éditeurs à la fois !) et recommencer.
- > Vérifier dans le transcript le succès de la simulation. En effet, le banc de test fourni est auto-vérifiant : il s'assure rapidement par des tests simples que le compteur semble fonctionner. C'est aussi une bonne idée de regarder dans le visualisateur des waveforms.

Écrire le décodeur 7-segments

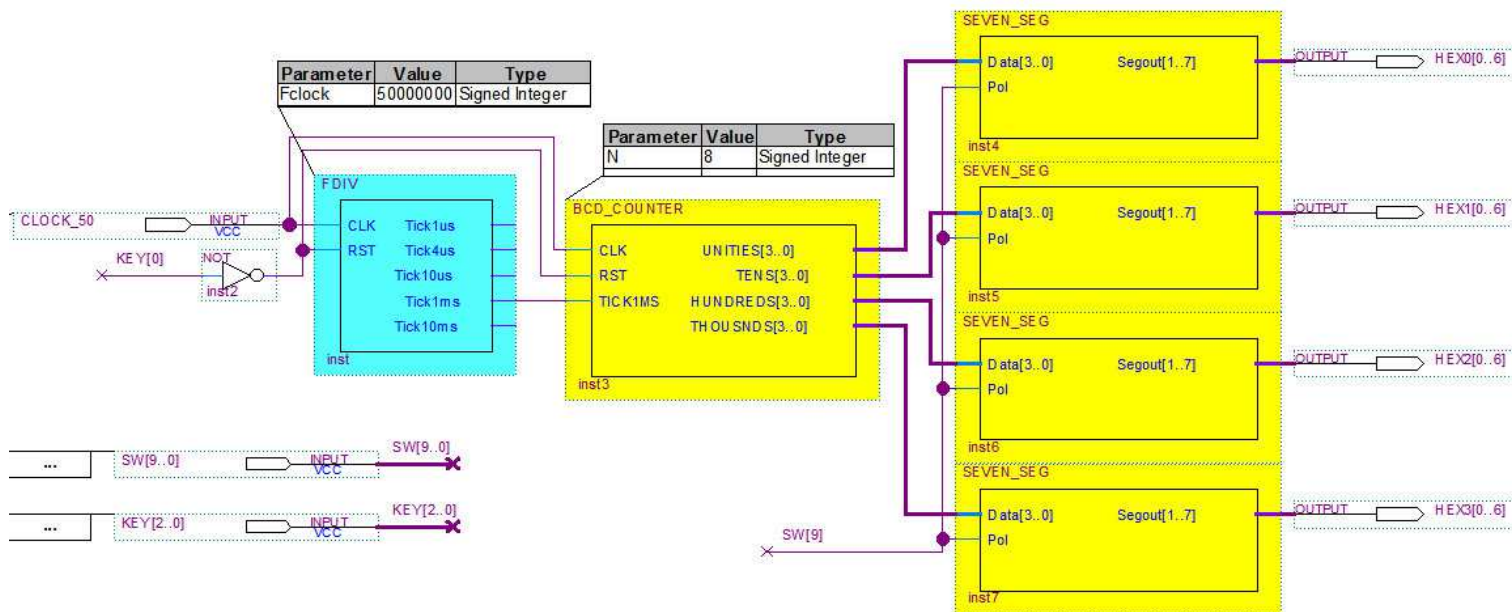
- > Ouvrir le fichier **Seven_Seg.vhd**.
- > Écrire le décodage en tenant compte des explications de l'énoncé (ne pas oublier la gestion de la polarité). Nous n'avons pas fourni de banc de test pour cette fonction car si vous deviez l'écrire, vous écririez certainement la même table dans le banc de test

que dans le décodeur, ce qui n'a (presque) aucune valeur au titre de vérification ! Il faut accepter dans ce cas la vérification visuelle du résultat.

- Noter que l'on ne précise rien pour les codes 10..15 (décimal), écartés par l'énoncé. Dans ce cas, il peut être intéressant d'optimiser la logique en laissant la synthèse adopter un code optimal pour ces cas non demandés (aide : il existe une des neuf valeurs `std_logic` qui est réservée à cet effet).

Écrire le module top level

- Ouvrir le fichier **TP1_BCD.vhd**.
- Compléter le module **TP1_BCD** qui permet d'instancier le composant **FDIV**, le composant **BCD_COUNTER** et 4 fois le module **SEVEN_SEG**. Comme sur le schéma ci-dessous :



Synthétiser et charger le composant programmable sur la maquette.

Créez un projet sur Quartus dans le répertoire **fit**, nommez le **TP1_BCD** sélectionnez le FPGA relatif à la carte de développement que vous utilisez (voir le « User Manuel » de cette carte). Rajouter vos sources en sélectionnant les sources qui sont dans le répertoire **src** (sans les copier dans le répertoire **fit**).

Sélectionnez le composant **TP1_BCD** comme entité de plus haute hiérarchie (top-level). Pour cela, sélectionnez le « File » **TP1_BCD.vhd** dans le Project Navigator, faites un clic droit avec la souris et sélectionnez « Set as top-level Entity ».

Maintenant il faut faire l'assignement des pins d'entrée/sortie :

Lancer une première fois la synthèse de votre projet. Ensuite assignez les pins du FPGA depuis le menu **Assignements** → **Pin Planner** en vous reportant au manuel de la carte pour choisir les bons numéros de pin sur la colonne **Location**.

Programmation du FPGA :

- Ouvrez le Programmeur qui se trouve dans Tools → Programmer.
- Connecter la carte de développement ,

- En haut à gauche de cette fenêtre, en face de Hardware Setup, si il n'y a pas noté USB_Blaster, cliquez sur Hardware Setup et choisissez USB_Blaster et cliquez sur Close.
- Lorsque USB_Blaster est bien sélectionné, cliquez sur Start et normalement le bitstream (fichier .sof) se charge dans le FPGA.
- La carte doit alors afficher le compteur rapide. Vérifier que l'affichage est correct, sinon corriger le module de décodage 7-segments et recommencer.

Amélioration :

- Pour l'instant, la vitesse d'incréméntation est fixée par un paramètre générique (N). Il faut recompiler le design à chaque fois que l'on veut modifier cette valeur.
- Dans cette partie, on propose de pouvoir régler la vitesse d'incréméntation du compteur à partir des switch SW0 à SW7. Ainsi, on pourra régler une vitesse d'incréméntation entre 1 et 256 ms. Il faudra incréménter le compteur après chaque « valeur entière de SW(7 downto 0) + 1 ».
- Pour éviter de toucher au code de la partie précédente, on propose de créer une nouvelle entité pour le module BCD_COUNTER (il faudra mettre la première entité/architecture en commentaire)

Vérification du travail réalisé :

Vous devrez faire vérifier le fonctionnement du compteur BCD à l'enseignant encadrant. Ensuite, vous noterez votre nom au tableau en face duquel vous noterez le nombre de Logic Element (LE) utilisé pour réaliser votre compteur BCD.

Le travail présenté doit être uniquement le travail de l'étudiant. Présenter le travail d'un autre sera sévèrement pénalisé.