



I. ASSEMBLAGE DES COMPOSANTS

Les fonctions principales sont maintenant réalisées, sauf la fonction FP6 mais cette fonction est disponible dans les bibliothèques Altium.

Le langage VHDL permet de gérer la hiérarchie d'un projet mais il est aussi possible de le faire plus facilement à l'aide d'une feuille de schéma, c'est cette solution que nous allons mettre en œuvre.

I.1. AJOUT D'UN NOUVEAU FICHIER SCHÉMA

Ajouter un nouveau schéma par un **clic droit** sur le nom du projet FPGA :

⇒ **Add New to Project** ⇒ **Schematic**

Puis sauvegarder ce fichier par un **clic droit** sur le nom du fichier Sheet1.SchDoc :

⇒ **Save As...** ⇒ **Top_Codeur_Incremental.SchDoc**

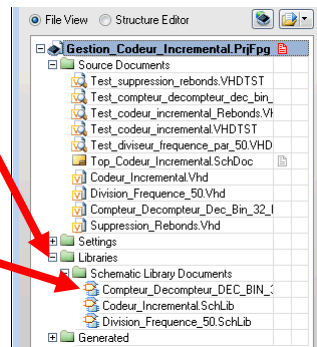
I.2. MODIFICATION DES SYMBOLES DES COMPOSANTS

Les symboles des composants ont été créés mais leur disposition n'est pas toujours judicieuse.

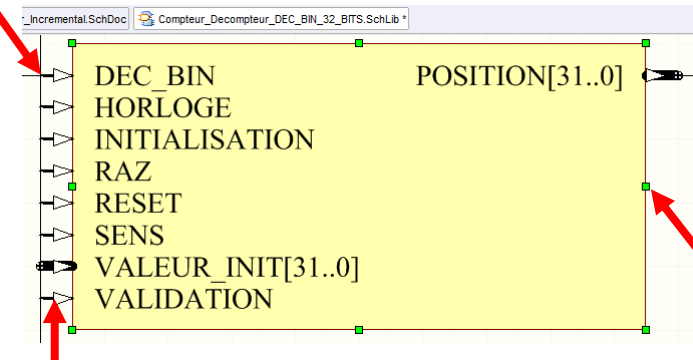
Faire apparaître les symboles du projet en cliquant sur les +
Du répertoire Bibliothèques.

Cliquer sur le symbole Compteur_Decompteur_...

Celui-ci apparaît dans une nouvelle fenêtre.



Le point d'ancrage du symbole est ici.

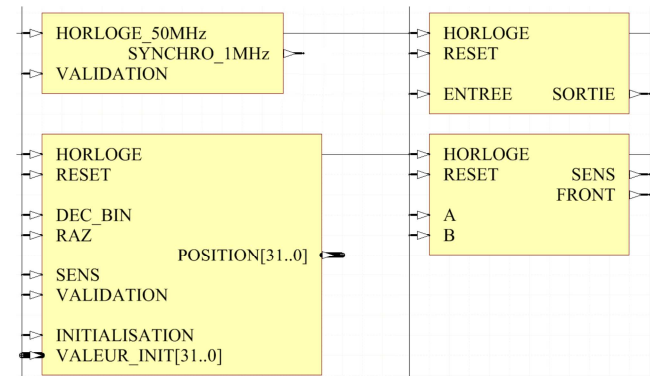


Les ports peuvent être déplacés de haut en bas en cliquant dessus

La taille du symbole peut être modifiée en cliquant sur le corps du symbole et en déplaçant les taquets.

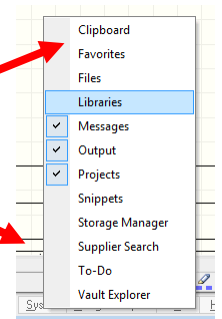
Enregistrer le symbole (CTRL + S) une fois modifié puis le fermer (CTRL + F4).

Proposition de disposition :

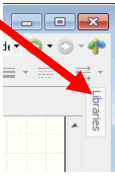


I.3. DESSIN DU SCHÉMA

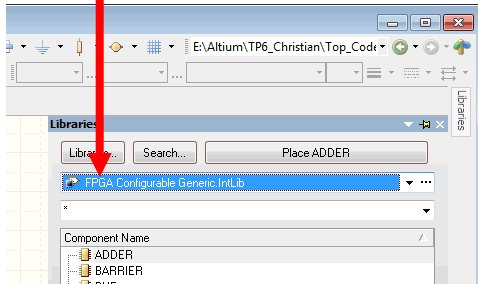
Vérifier que l'onglet **Libraries** est bien présent en cliquant sur System en bas à droite de la fenêtre Altium.



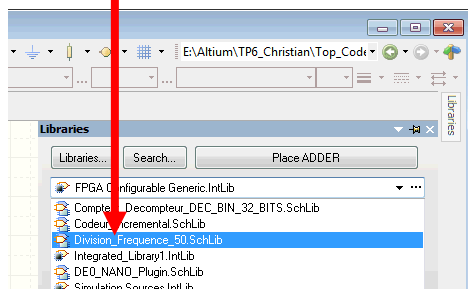
Pour placer un symbole sur la feuille de schéma :
Passer la souris sur l'onglet **Libraries** :



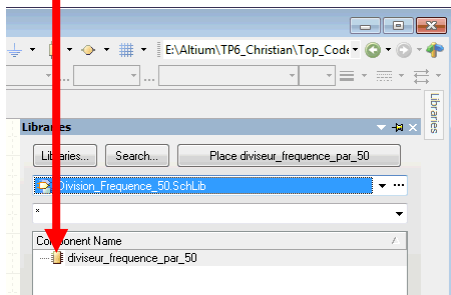
L'onglet **Libraries** apparaît, cliquer sur le menu déroulant de sélection des bibliothèques :



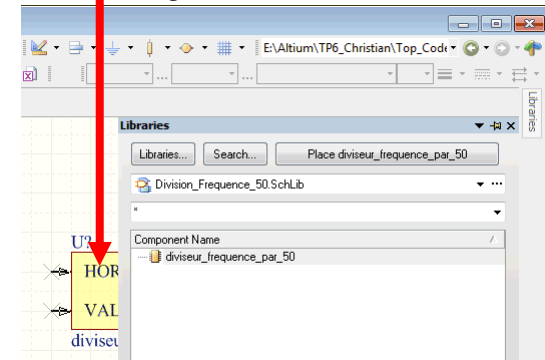
Sélectionner **Division_Frequency_50.SchLib** en cliquant dessus :



Cliquer sur le composant : **Division_Frequency_50**



Et effectuer un glisser-coller du composant sur la feuille de schéma.

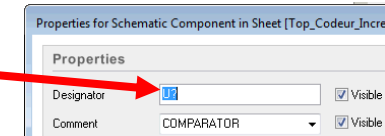


Placer les différents composants comme le schéma donné en annexe page 9.

Le comparateur (FP6) est placé dans la librairie : **FPGA Configurable Generic.IntLib** et se nomme **COMPARATOR**.

Une fois placé sur le schéma, configurer celui-ci en double-cliquant dessus puis :

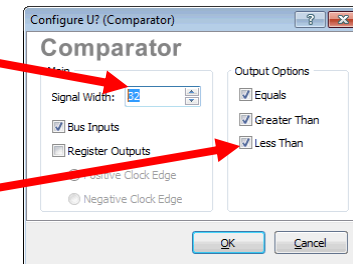
Le renommer en U6



Ouvrir la fenêtre de configuration



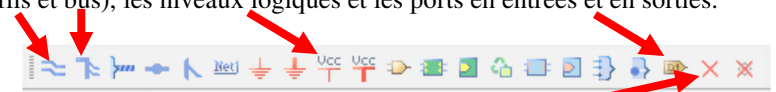
Configurer la largeur du bus



Choisir les sorties

Renommer chaque composant U1 pour FP1, etc... En double-cliquant sur chaque composant.

Placer les connexions (fils et bus), les niveaux logiques et les ports en entrées et en sorties.



Si une sortie n'est pas utilisée il convient de placer un terminateur.

Lorsque le schéma est terminé sauvegarder celui-ci puis le compiler. Corriger les éventuelles erreurs.

II. DESSIN DU SCHÉMA TOP

Altium permet de gérer la hiérarchie des schémas : un schéma de haut niveau peut contenir un ou plusieurs autres schémas, etc...

II.1. AJOUT D'UN NOUVEAU FICHIER SCHÉMA

Ajouter un nouveau schéma par un **clic droit** sur le nom du projet FPGA :

⇒ **Add New to Project** ⇒ **Schematic**

Puis sauvegarder ce fichier par un **clic droit** sur le nom du fichier Sheet1.SchDoc :

⇒ **Save As...** ⇒ **Gestion_Codeur_Incremental.SchDoc**

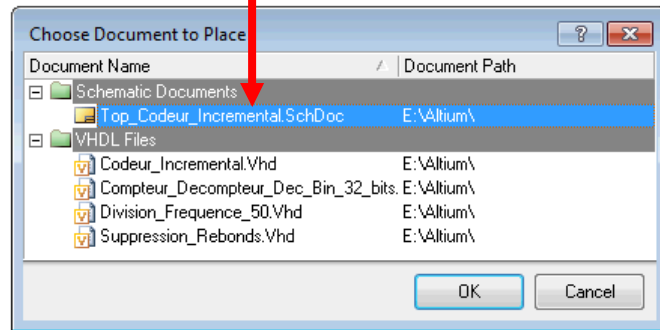
Remarque : le schéma top doit obligatoirement avoir le même nom que le projet.

II.2. PLACEMENT D'UN SCHÉMA DANS UN SCHÉMA SUPÉRIEUR

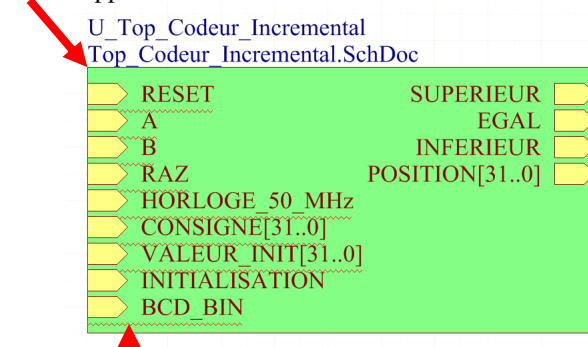
Se placer sur le schéma supérieur (TOP) puis faire :

⇒ **Design** ⇒ **Create Sheet Symbol from Sheet or HDL**

Puis sélectionner Top_Codeur_Incremental.SchDoc

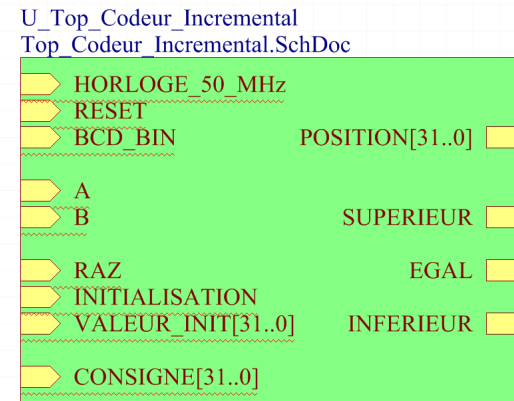


Un symbole de schéma apparaît au bout de la souris. Placer celui-ci sur la feuille de schéma.

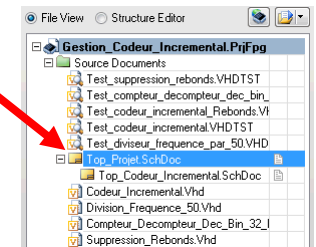


Remarque : Indique qu'une entrée est en l'air.

Afin d'obtenir un symbole mieux structuré il est possible de modifier la taille et de déplacer les ports en entrées et en sorties.



On peut remarquer dans l'onglet Projet la hiérarchie des schémas. C'est bien **Top_Projet.SchDoc** qui est le top du projet.



II.3. SCHÉMA TOP LEVEL

Ajouter des ports d'entrées et de sorties comme sur le schéma donné en annexe page 10.

La sortie EGALITE n'est pas utilisée, placer un terminateur.

Ajouter deux connecteurs J4B8_32B : ils permettent d'assembler 8 bus de 4 bits en un bus de 32 bits. Ils se trouvent dans la librairie **FPGA Generic.IntLib**.

Ajouter les composants NUMC0, NUMC1, NUMC2, NUMC5 qui se trouvent dans la librairie **FPGA Generic.IntLib**. Ces composants permettent de fixer la valeur d'un mot de 4 bits.

NUMC5 fourni le mot binaire 0101 par exemple.

VALEUR_INIT = 00005125

CONSIGNE = 00005250

Numéroter les composants avec la commande :

⇒ **Tools** ⇒ **Annotate Schematics Quietly...**

Sauvegarder le schéma puis le compiler.

Sauvegarder le projet puis le compiler.

III. SIMULATION FONCTIONNELLE

III.1. PRODUCTION DU FICHIER DE TEST

Créer un fichier VHDL de test à partir du schéma TOP du projet avec :

⇒ **Simulator** ⇒ **Create VHDL Testbench**

Sauvegarder le nouveau fichier de tests avec le nom proposé.

Compléter celui-ci avec l'extrait donné en annexe page 10.

III.2. ASSOCIATION DES FICHIERS

Ajouter le fichier de test dans le Manager Testbenches du simulateur.

III.3. SIMULATION

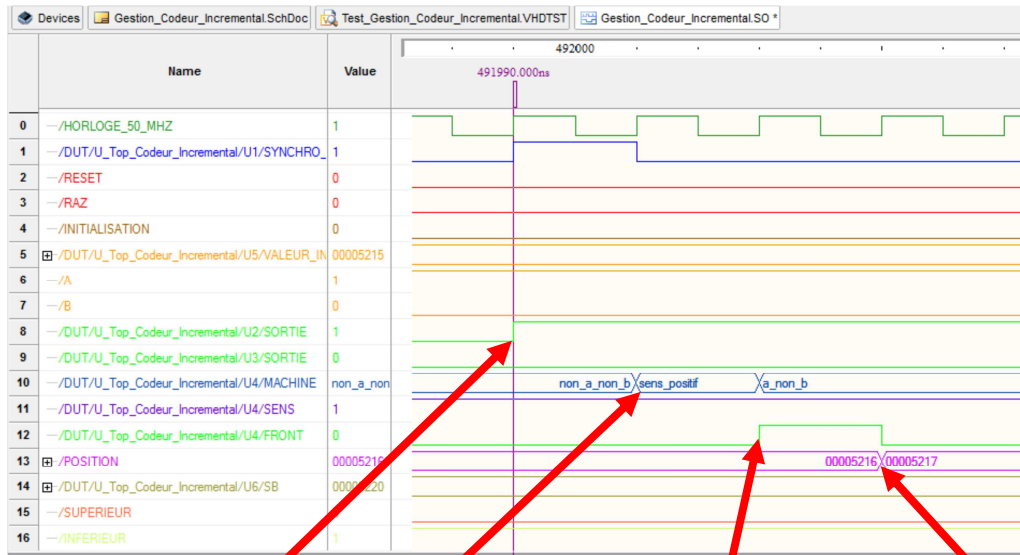
Lancer la simulation.

Remarque : L'horloge HORLOGE_50_MHz a une période de 20 ns. Le signal HORLOGE à une période de 1µs (division par 50).

La fonction de suppression des rebonds génère un retard de 34 coups d'horloge ce qui donne 34 µs, la période du signal A ou B doit être au moins 2 fois plus grande. Pour la simulation la période a été fixée à 204 µs.

Ajouter quelques signaux internes comme sur le chronogramme suivant :





Détection d'un front Traitement Ordre de comptage positif Evolution du compteur

Le projet complet est validé.

IV. IMPLANTATION DANS UN FPGA

IV.1. SCHÉMA A COMPLÉTER

Afin de pouvoir programmer le FPGA de la carte DE2-115 de Terasic il faut ajouter 8 décodeurs binaire/7 segments à anodes communes afin de pouvoir visualiser la valeur du compteur POSITION sur les afficheurs 7 segments de la carte.

Il faut aussi ajouter un connecteur de bus et des inverseurs car les boutons-poussoirs de la carte Terasic sont actifs à l'état bas.

Le décodeur se nomme **D7SEGNB**, le connecteur se nomme **J32B_4B8X** et l'inverseur se nomme **INV**. Ils sont tous placés dans la librairie : **FPGA Generic.IntLib**.

Terminer le schéma comme donné en annexe page 11. Puis sauvegardez celui-ci avant de le compiler.

IV.2. COMPILATION DE L'ENSEMBLE DU PROJET

Effectuer une sauvegarde et une compilation de l'ensemble du projet en cliquant droit sur le nom du projet et en choisissant **Save Project** puis **Recompile FPGA Project ...**

A ce stade il ne doit pas rester d'erreurs.

IV.3. FICHIER DE CONTRAINTES

Un fichier de contraintes au format Altium doit être produit. Celui-ci contient un certain nombre d'informations indispensables pour filtrer le projet dans un composant réel comme le type de composant ou le brochage des signaux d'entrées et de sorties.

Pour créer ce fichier cliquer droit sur le nom du projet puis :

⇒ **Add New to Project** ⇒ **Constraint File**

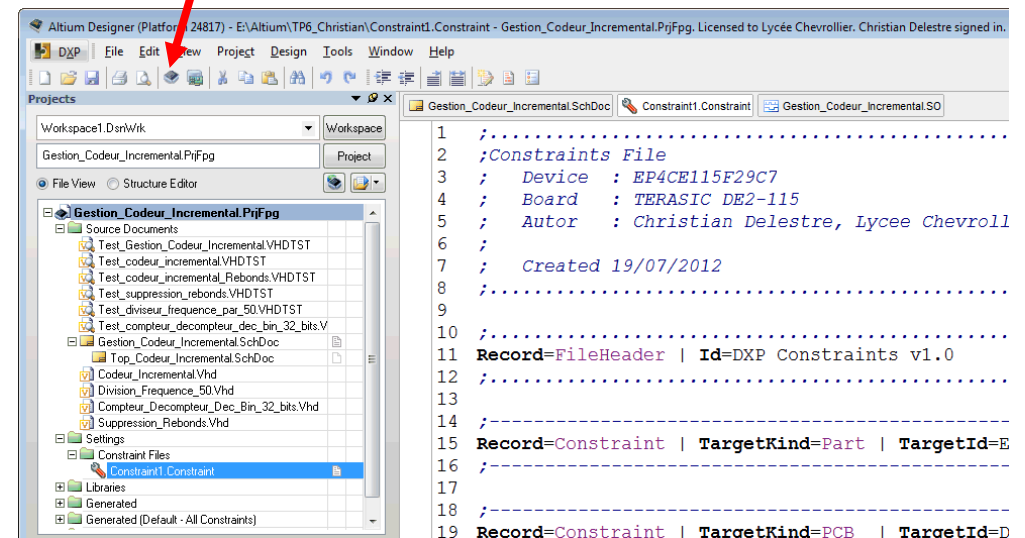
Sauvegarder ce fichier en gardant le nom proposé.

Compléter celui-ci comme indiqué en annexe page 12.

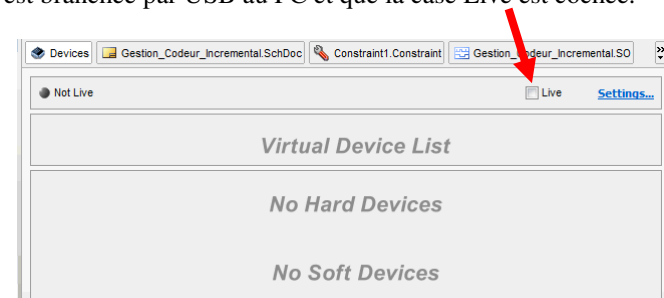
IV.4. PRÉPARATION AVANT CONSTRUCTION DU PROJET COMPLET

Sélection du composant cible.

Cliquer sur l'icône **Open the Device View Page**.



Avec une carte NANOBOARD 3000 d'Altium la reconnaissance se fait automatiquement lorsque celle-ci est branchée par USB au PC et que la case Live est cochée.

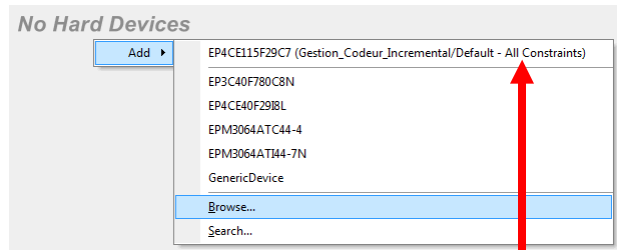


Lorsqu'on utilise une autre carte, non compatible avec Altium il convient d'effectuer une configuration manuelle.

Effectuer un clic droit dans la fenêtre au niveau du texte **No Hard Device**.

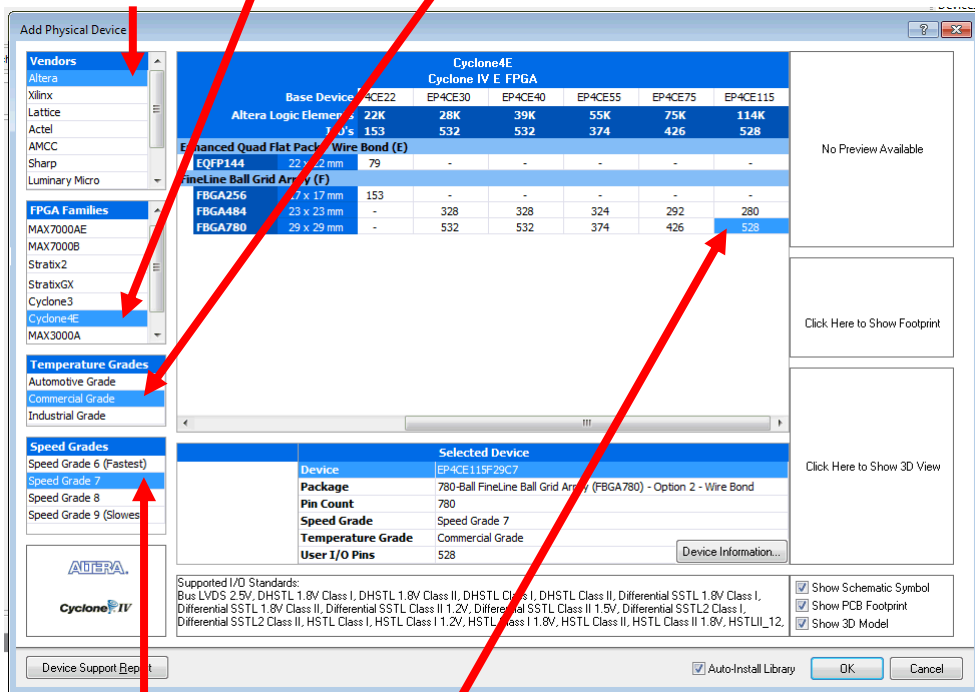


Puis déplacer la souris sur **Add**



Si le fichier de contrainte a bien été pris en compte la première ligne apparaît, cliquer dessus. Sinon cliquer sur **Browse...** afin de sélectionner le composant cible.

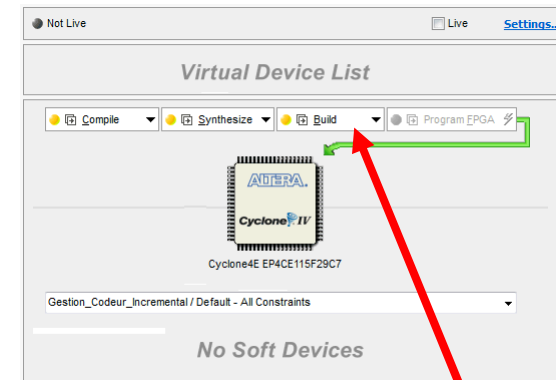
Sélectionner **Altera** puis **Cyclone4E** puis **Commercial Grade**



Puis **Speed Grade 7** puis **EPC4CE115 – 528** et enfin **OK**.

IV.5. CONSTRUCTION DU PROJET COMPLET

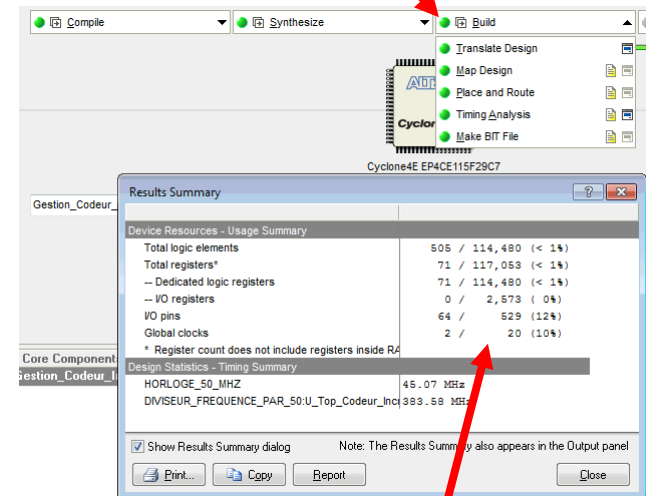
Le composant apparaît dans la fenêtre :



Il ne reste plus qu'à construire le projet complet en cliquant sur **Build**.

Le projet se construit doucement. Remarquer qu'Altium utilise Quartus (Alera) pour construire le projet.

A la fin toutes les étapes doivent être passées au vert :



Un résumé permet d'avoir une vision globale du résultat.

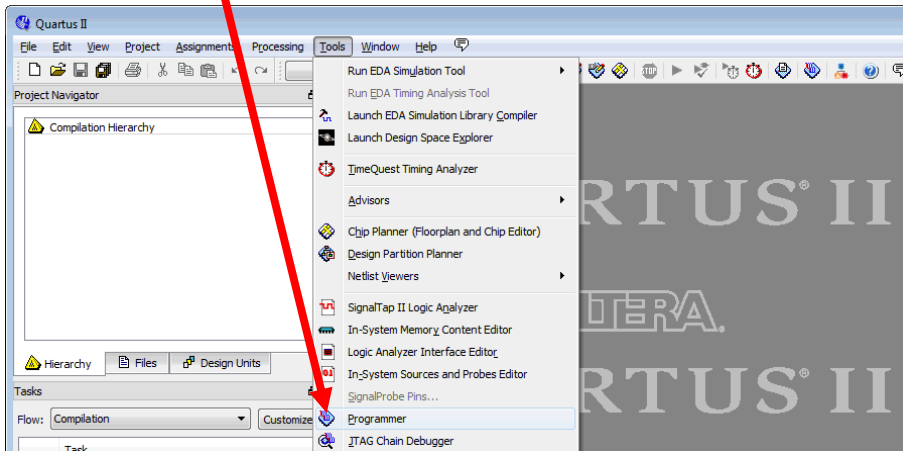
V.6. PROGRAMMATION DU COMPOSANT

Si la carte cible est une NANOBOARD3000 il suffit de cliquer sur le bouton **Program FPGA**.

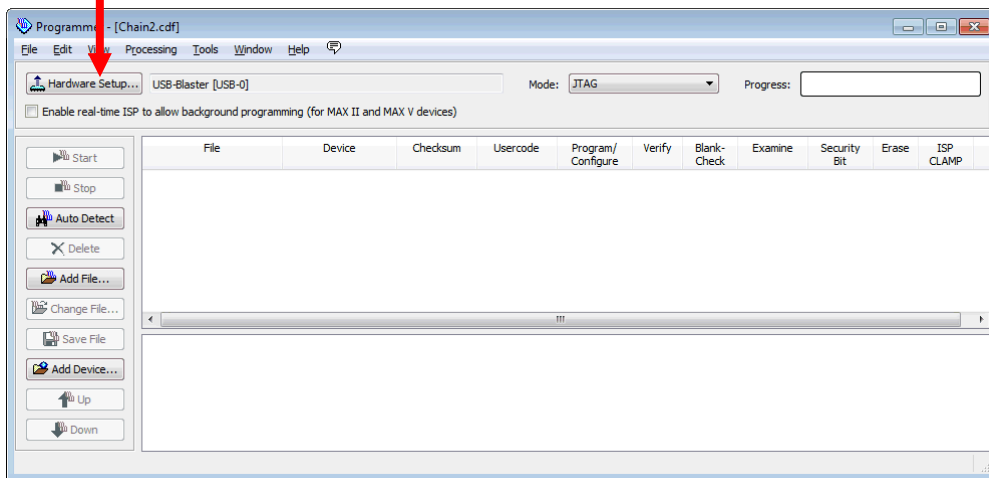
Pour une autre carte il faut utiliser l'outil de programmation du FPGA.

Dans le cas de la carte DE2-115 de Terasic il faut lancer QUARTUS d'ALTERA.

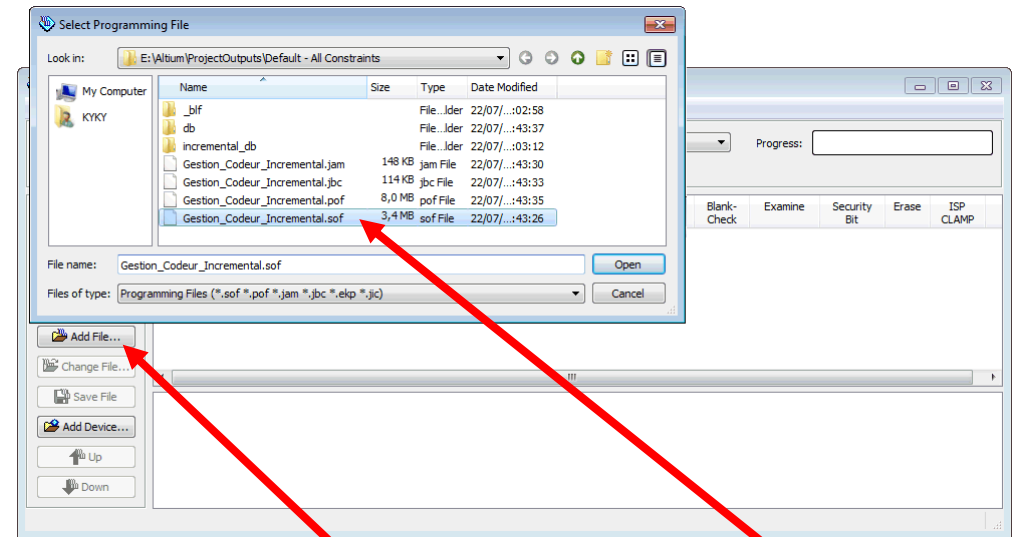
Lancer le programmeur à partir du menu Tools :



Sélectionner le programmeur USB-Blaster.

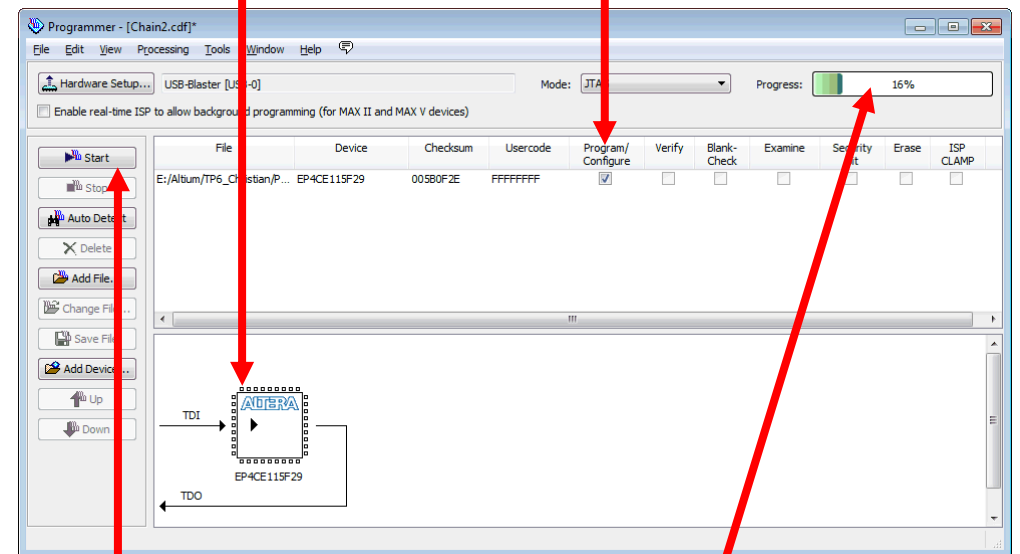


Remarque : dans cet exemple nous considérerons que l'installation des drivers USB-Blaster d'Altera est faite correctement.



Ajouter un fichier en cliquant sur **Add File...** Puis à partir de la fenêtre de sélection du fichier aller dans le répertoire du projet, puis dans **ProjectOutputs** puis dans **Default - All Constraints**. Sélectionner le fichier **Gestion_Codeur_Incremental.sof**.

Le composant apparaît. Cocher la case Program/Configure.



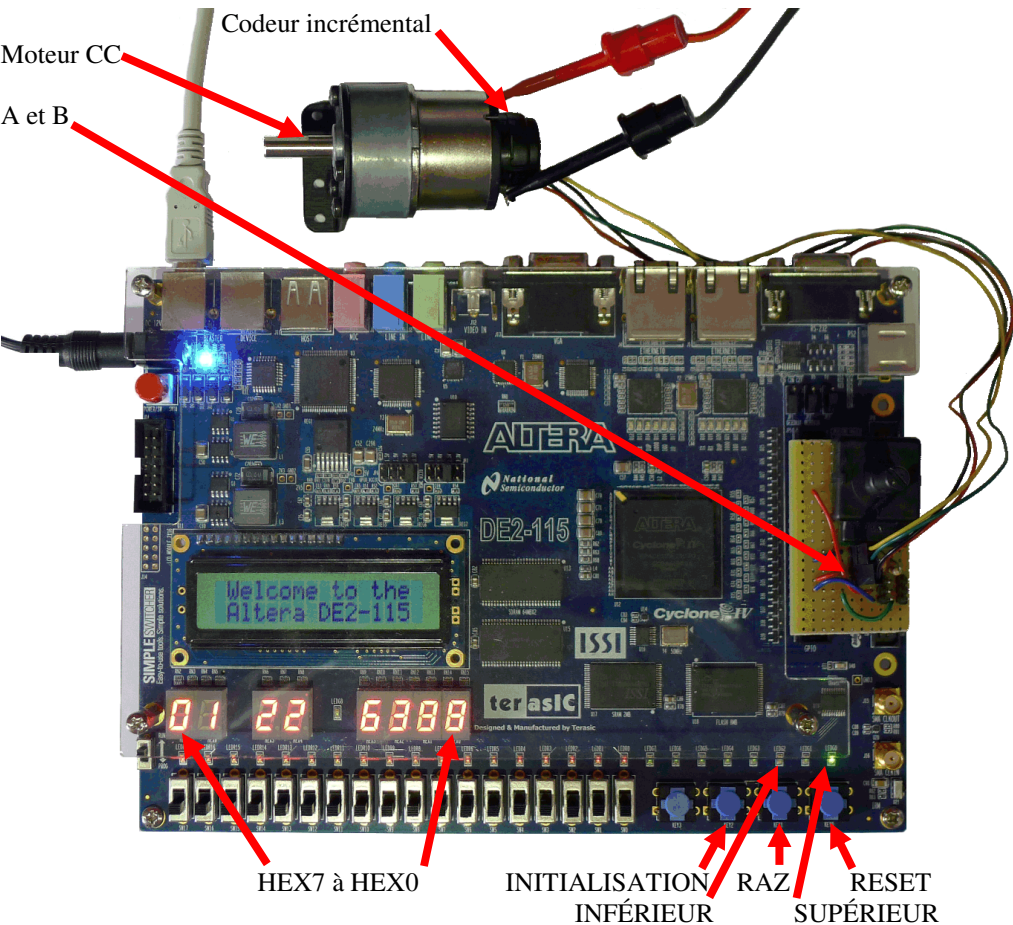
Cliquez sur **Start**. La programmation doit se dérouler jusqu'à 100%. Le composant FPGA est programmé.

VI. TESTS DE VALIDATION MATERIELLE

VI.1. MISE EN ŒUVRE

Pour les tests une petite carte de prototypage permet d'utiliser soit un codeur incrémental de tableau (manuel) ou un codeur incrémental disposé en bout d'axe d'un moteur à courant continu. C'est avec ce dernier que les mesures seront faites.

Le fichier de contraintes a défini les broches comme indiqué ci-dessous :



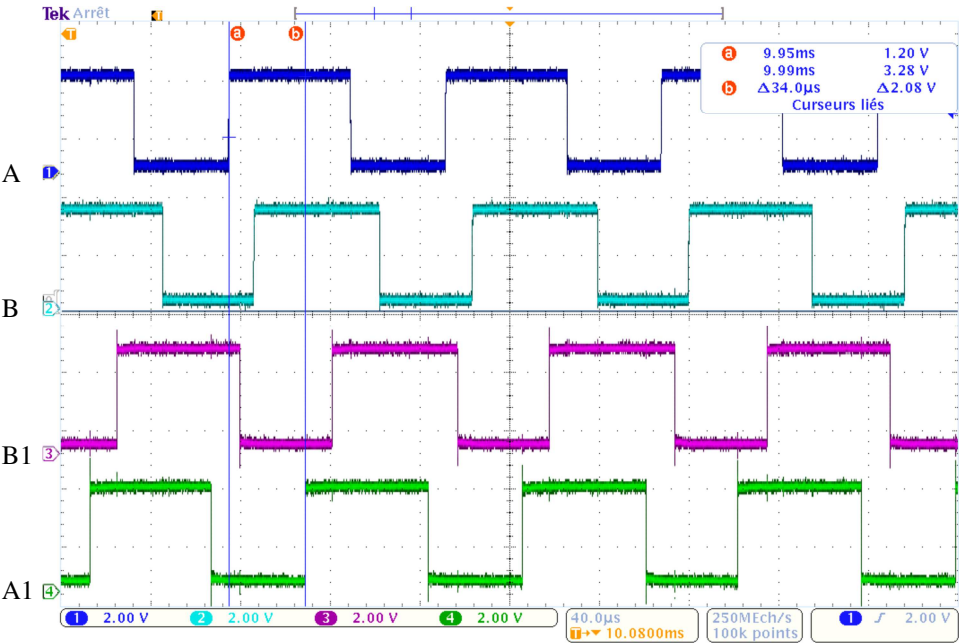
On peut remarquer que lorsque le moteur tourne l'affichage des unités, dizaines et centaines n'est pas lisible du fait de la fréquence élevée d'évolution des chiffres.

Liste des composants utilisés :

Désignation	Référence	Fournisseur
Carte de développement DE2-115	P0103-ND	DIGI-KEY
Motoréducteur GHM16	01601	GOTORONIC
Paire de supports pour GHM16	01598	GOTORONIC
Encodeur pour moteur GHM16	01599	GOTORONIC
Codeur incrémental ECW0J-B24-AC0006L	2292941	FARNELL
Equerre de fixation	219113	FARNELL

VI.2. VALIDATION

Un chronogramme a été pris en sortant les points A, A1, B, B1.



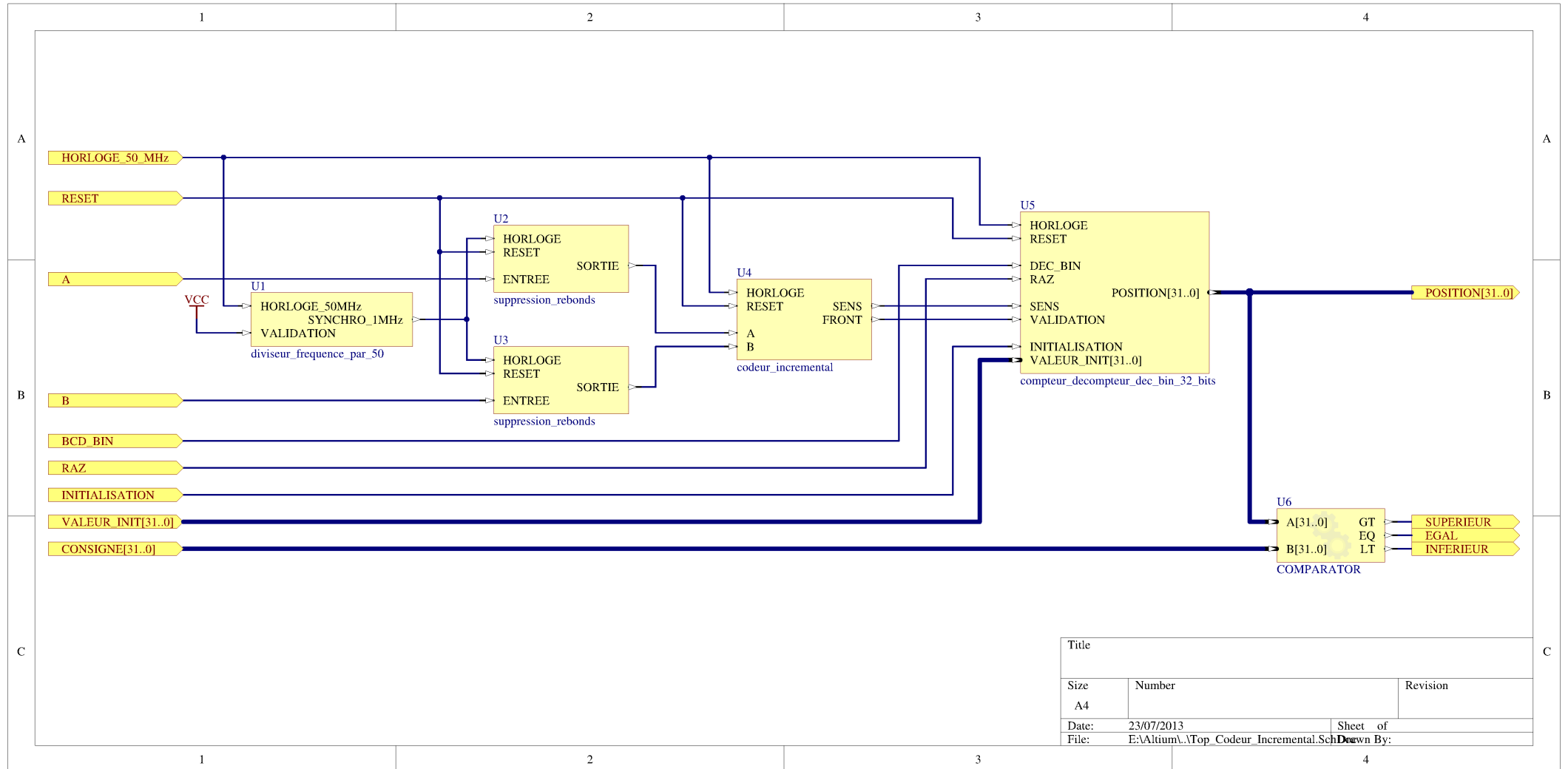
On retrouve le retard de 34 µs entre la sortie et l'entrée des fonctions FP2 et FP3. On peut remarquer qu'avec le moteur alimenté en 12 V la fréquence des signaux A et B est d'environ 10 kHz ce qui fait un front à traiter toutes les 25 µs.

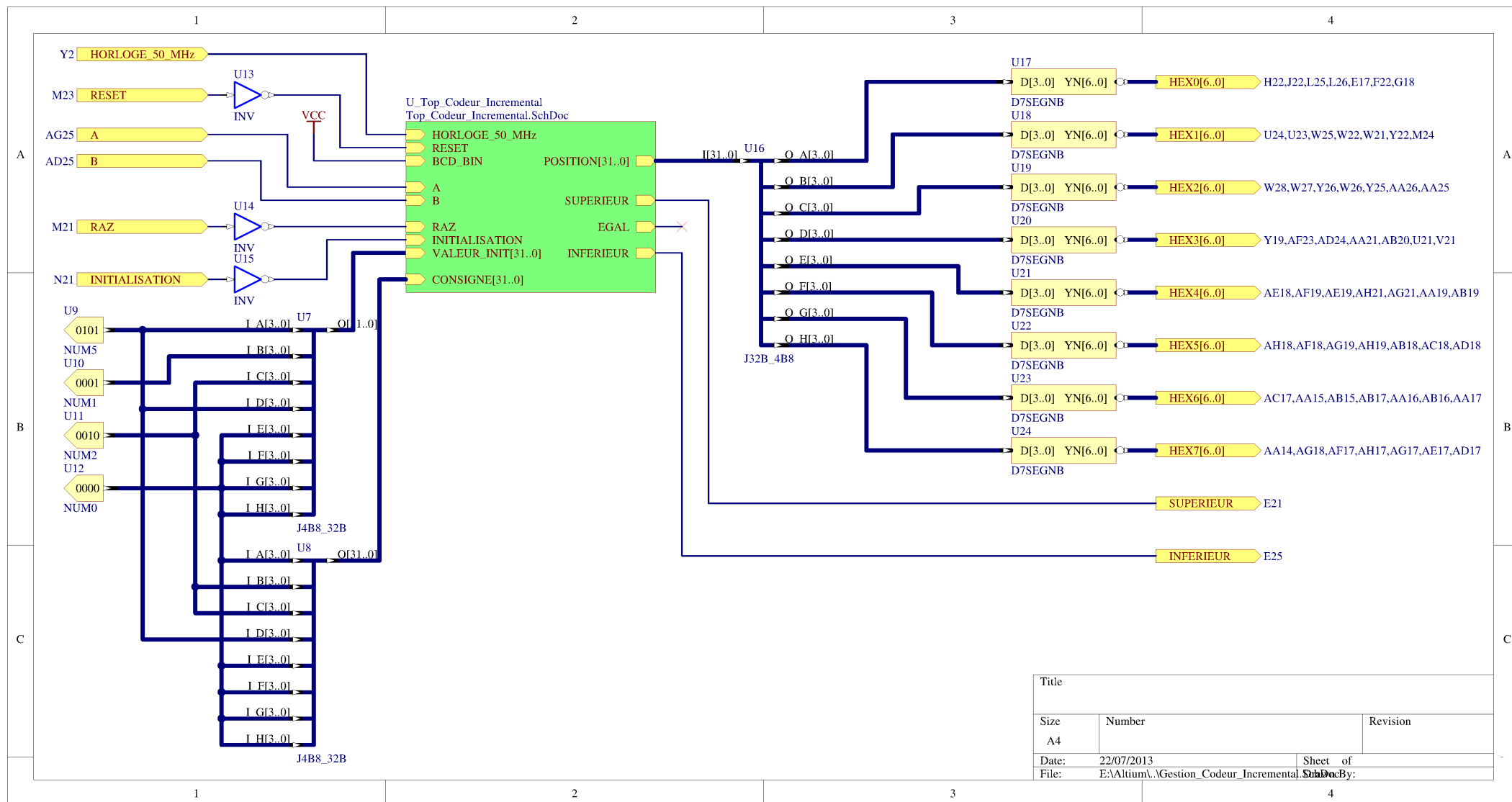
L'action des 2 boutons et l'évolution de l'affichage en fonction de la vitesse de rotation du moteur, ainsi que son sens de rotation permettent de valider le fonctionnement de l'ensemble.

Le projet de gestion d'un codeur incrémental est validé.

Utilisation possible : dans un robot, un capteur sur chaque roue afin de contrôler son déplacement. Dans ce cas il suffit de placer 2 sous-schémas dans le schéma top.

VII. ANNEXES





```

1  ;.....
2  ;Constraints File
3  ;   Device   : EP4CE115F29C7
4  ;   Board    : TERASIC DE2-115
5  ;   Autor    : Christian Delestre, Lycee Chevroliier
6  ;
7  ;   Created  19/07/2012
8  ;.....
9
10 ;.....
11 Record=FileHeader | Id=DXP Constraints v1.0
12 ;.....
13
14 ;-----
15 Record=Constraint | TargetKind=Part | TargetId=EP4CE115F29C7
16 ;-----
17
18 ;-----
19 Record=Constraint | TargetKind=PCB   | TargetId=DE2-115      | Image=DE2-115
20 ;-----
21
22 Record=Constraint | TargetKind=Port | TargetId=HORLOGE_50_MHz | FPGA_PINNUM=Y2
23
24 Record=Constraint | TargetKind=Port | TargetId=RESET          | FPGA_PINNUM=M23
25 Record=Constraint | TargetKind=Port | TargetId=RAZ             | FPGA_PINNUM=M21
26 Record=Constraint | TargetKind=Port | TargetId=INITIALISATION | FPGA_PINNUM=N21
27
28 Record=Constraint | TargetKind=Port | TargetId=A              | FPGA_PINNUM=AG25
29 Record=Constraint | TargetKind=Port | TargetId=B              | FPGA_PINNUM=AD25
30
31 Record=Constraint | TargetKind=Port | TargetId=HEX0[6..0]     | FPGA_PINNUM=H22,J22,L25,L26,E17,F22,G18
32 Record=Constraint | TargetKind=Port | TargetId=HEX1[6..0]     | FPGA_PINNUM=U24,U23,W25,W22,W21,Y22,M24
33 Record=Constraint | TargetKind=Port | TargetId=HEX2[6..0]     | FPGA_PINNUM=W28,W27,Y26,W26,Y25,AA26,AA25
34 Record=Constraint | TargetKind=Port | TargetId=HEX3[6..0]     | FPGA_PINNUM=Y19,AF23,AD24,AA21,AB20,U21,V21
35 Record=Constraint | TargetKind=Port | TargetId=HEX4[6..0]     | FPGA_PINNUM=AE18,AF19,AE19,AH21,AG21,AA19,AB19
36 Record=Constraint | TargetKind=Port | TargetId=HEX5[6..0]     | FPGA_PINNUM=AH18,AF18,AG19,AH19,AB18,AC18,AD18
37 Record=Constraint | TargetKind=Port | TargetId=HEX6[6..0]     | FPGA_PINNUM=AC17,AA15,AB15,AB17,AA16,AB16,AA17
38 Record=Constraint | TargetKind=Port | TargetId=HEX7[6..0]     | FPGA_PINNUM=AA14,AG18,AF17,AH17,AG17,AE17,AD17
39
40 Record=Constraint | TargetKind=Port | TargetId=SUPERIEUR      | FPGA_PINNUM=E21
41 Record=Constraint | TargetKind=Port | TargetId=INFERIEUR      | FPGA_PINNUM=E25

```