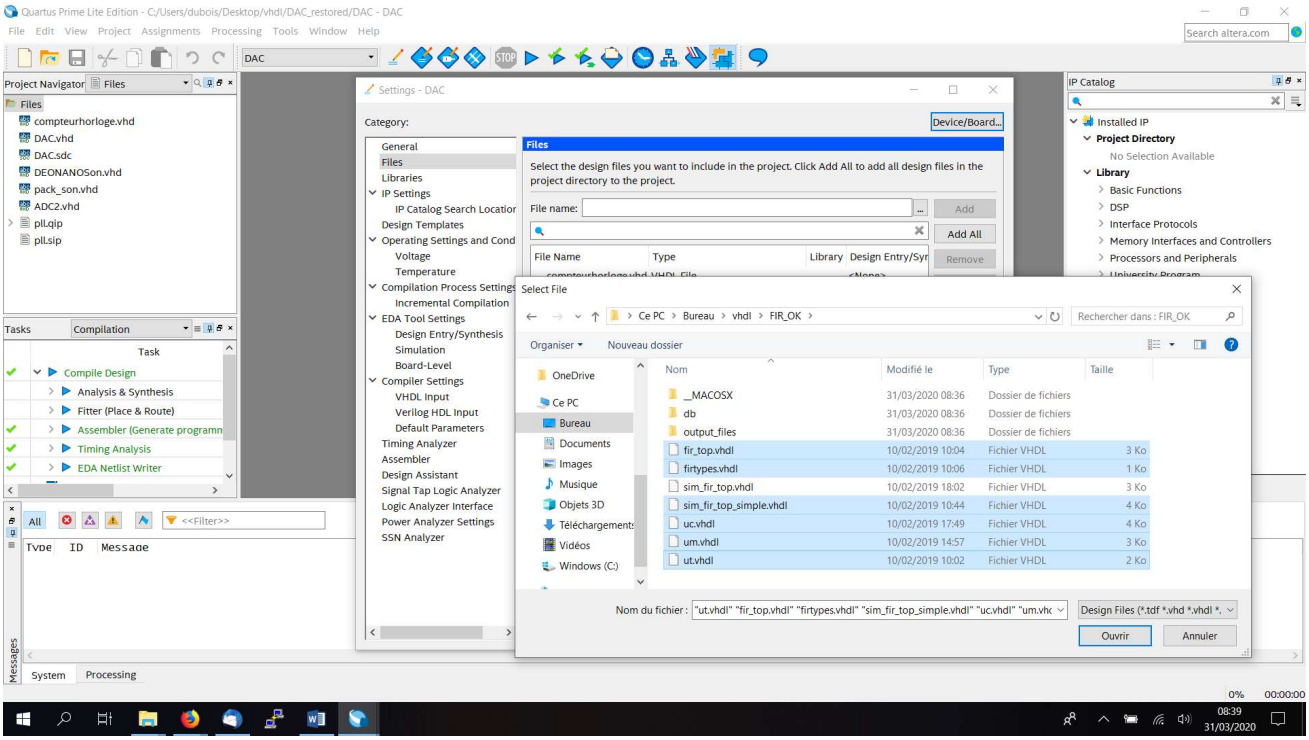
Tutorial Quartus Modelsim « confiné » projet VHDL ELEC2 S2 2020 sur l'exemple du FIR.

Vous avez utilisé jusqu'alors l'environnement Synopsys pour faire la synthèse logique des codes VHDL. Pour le projet, vous utiliserez un outil propriétaire d'Altera : Quartus. Vous avez déjà utilisé Quartus en première année. C'est un environnement complet qui permettra la saisie des codes VHDL, la synthèse logique, le placement routage et la programmation du FPGA.
Le but de ce tutorial est de vous guider sur la version Quartus Prime de ce logiciel au travers de l'exemple du FIR vu précédemment, et de son lien avec Modelsim. Il vous donne également des exemples de code structurés qui peuvent vous servir de base pour l'écriture de vos codes. Il vous donne enfin une méthodologie de développement et des conseils à appliquer pour le projet.
Vous pouvez utiliser les fichiers du FIR séquentiel sur Moodle, ou vos propres fichiers du FIR.

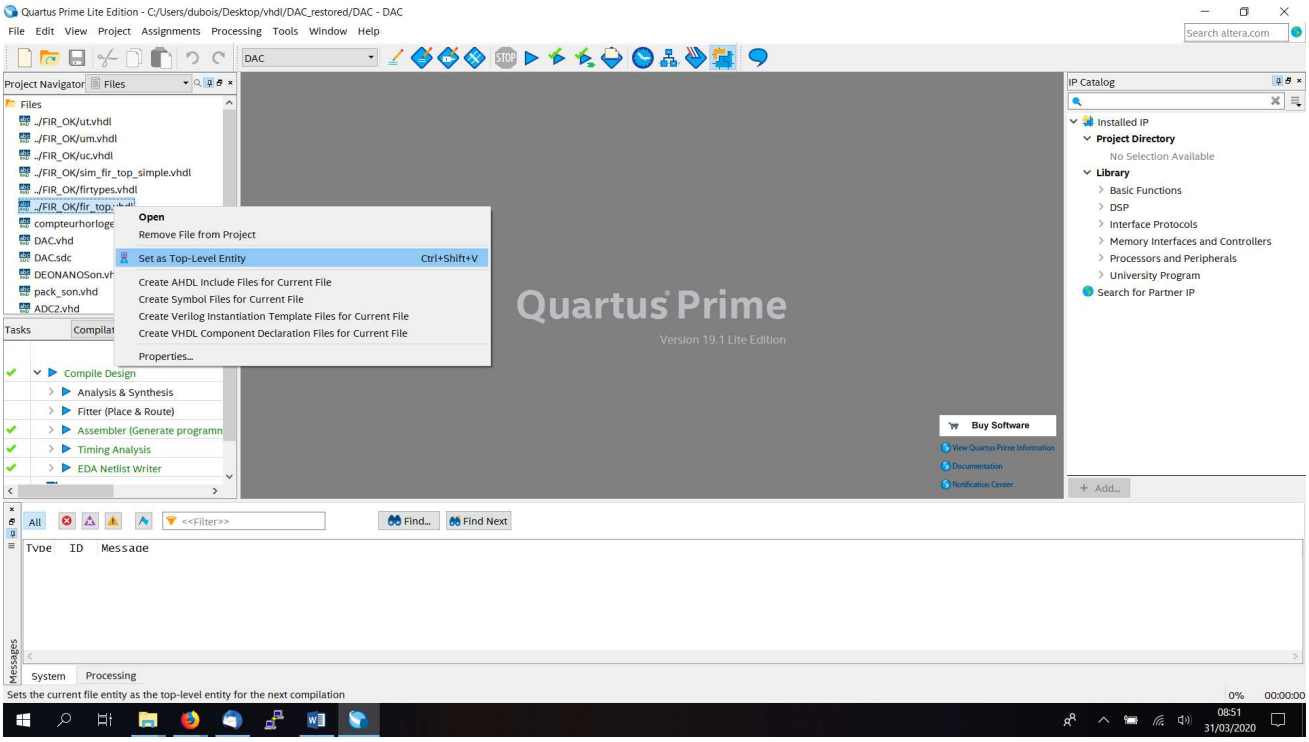
1 Intégration de nouveaux fichiers dans un projet existant et top level entity.

Quartus travaille avec la notion de projet, un projet étant constitué d'un ensemble de fichiers mais aussi d'un ensemble de configurations pour paramétrer le projet (choisir le composant ciblé par exemple, le placement des entrées sorties sur les broches du composant, etc).
Ici vous avez déjà le projet de départ avec la V0, et nous allons voir comment insérer le FIR dans ce projet. Cela évite d'avoir plusieurs projets qui occupent pas mal de place disque, et vous pourrez utiliser cette méthode pour développer vos différentes versions du projet.
Ouvrez le projet DAC grâce au menu File / Open project ou en double cliquant sur le fichier DAC.qpf (Quartus Project File). Insérez ensuite tous les fichiers nécessaires par le menu Project/ Add Remove Files to project.



Ensuite par le menu Assignement/ Settings , choisissez « Modelsim Altera» (et pas Modelsim) comme outil de simulation dans la fenêtre « EDA Tools Settings » .
Les paramètres entrés via des fenêtres peuvent à tout moment être visualisés et éventuellement modifiés via le menu « Assignements/ Settings ».

Basculez maintenant « fir_top » comme « top level entity », c'est-à-dire l'entité qui sera la plus haute dans la hiérarchie du projet. Pour cela, dans la fenêtre « Project Navigator » en haut à gauche, changez l'affichage « Hierarchy » vers l'affichage « Files », puis grâce à un clic droit sur le fichier ciblé dans cette fenêtre, sélectionnez « Set as Top level entity ».



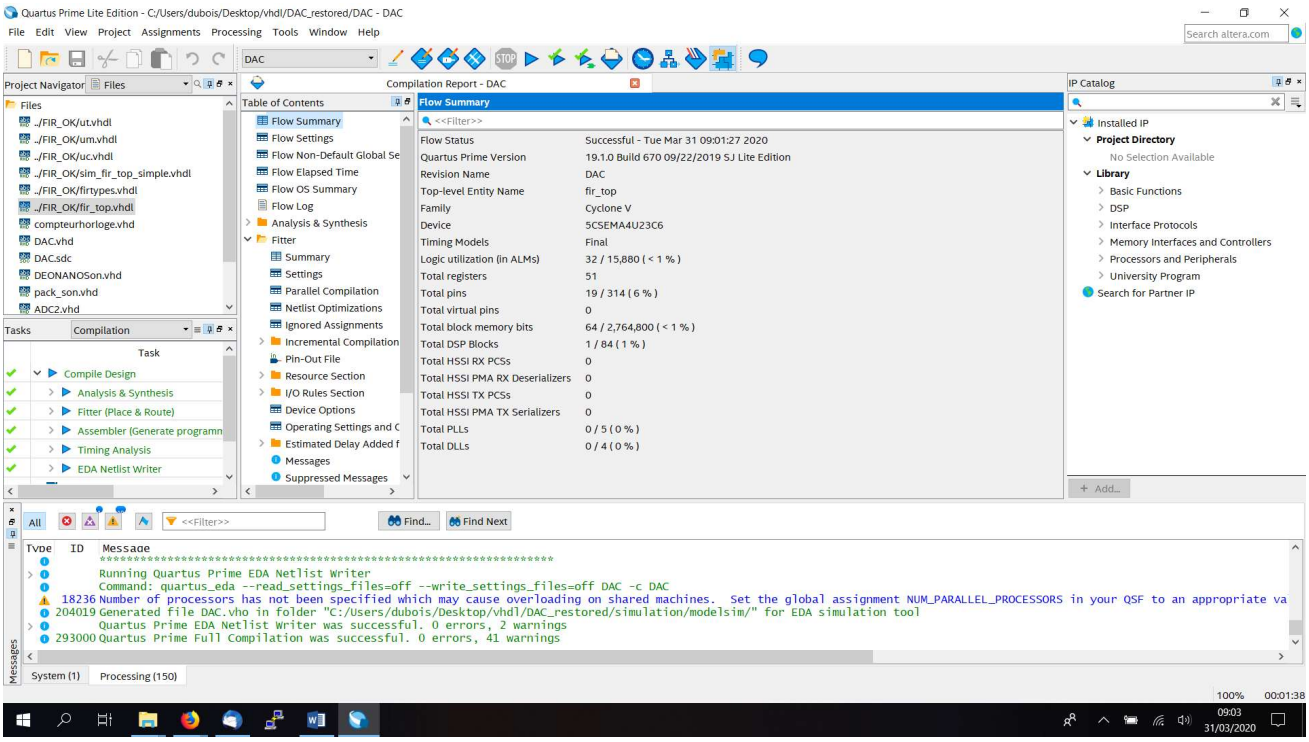
Le projet pointe donc maintenant sur le Fir au lieu de la V0. Les fichiers de la V0 sont quand même restés dans le projet et seront analysés mais ils ne seront pas intégrés dans le FPGA puisque la top level entity n'y a pas de lien. Vous pouvez retirer les fichiers de la V0 pour gagner un peu de temps à la compilation, mais vous pouvez aussi les laisser. Vous aurez des warning lors de la compilation par exemple sur le fait que le placement des broches est ignoré, ce qui est normal puisque les broches du FIR ne correspondent pas aux broches de la V0, mais ce n'est pas grave tant qu'on ne souhaite pas faire tourner le FIR seul sur le FPGA.

NB : si vous souhaitez compiler et faire la synthèse d'un autre fichier (par exemple l'UC seule) sans recréer un autre projet (car cela prend de la place sur le disque), vous pouvez changer la « top level entity » Attention : quand vous lancerez la compilation, c'est sur ce fichier qu'elle sera lancée et pas sur le fichier édité à l'écran.

2 Compilation et analyse.

Lancez la compilation grâce au menu « Processing/Start Compilation » ou à l'icône correspondant dans la barre d'outils.
Cela va lancer toutes les étapes nécessaires à aboutir à la création de tous les fichiers pour la programmation du composant, la simulation et l'analyse de la synthèse et du placement routage. L'avancée de ces étapes peut être vue dans la fenêtre « Tasks ». NB : si vous ne souhaitez lancer que les premières étapes, vous pouvez double cliquer sur l'étape à atteindre dans cette fenêtre (par exemple RTL Viewer). Cela permet de ne pas lancer des étapes non nécessaires dans un premier temps.
Si vous avez une erreur dans vos codes, un message est affiché dans la fenêtre « Message » et en double-cliquant sur le message, le fichier où se trouve l'erreur est ouvert sur la ligne posant problème. Commencez toujours par chercher la première erreur, car une erreur sur une ligne peut en provoquer

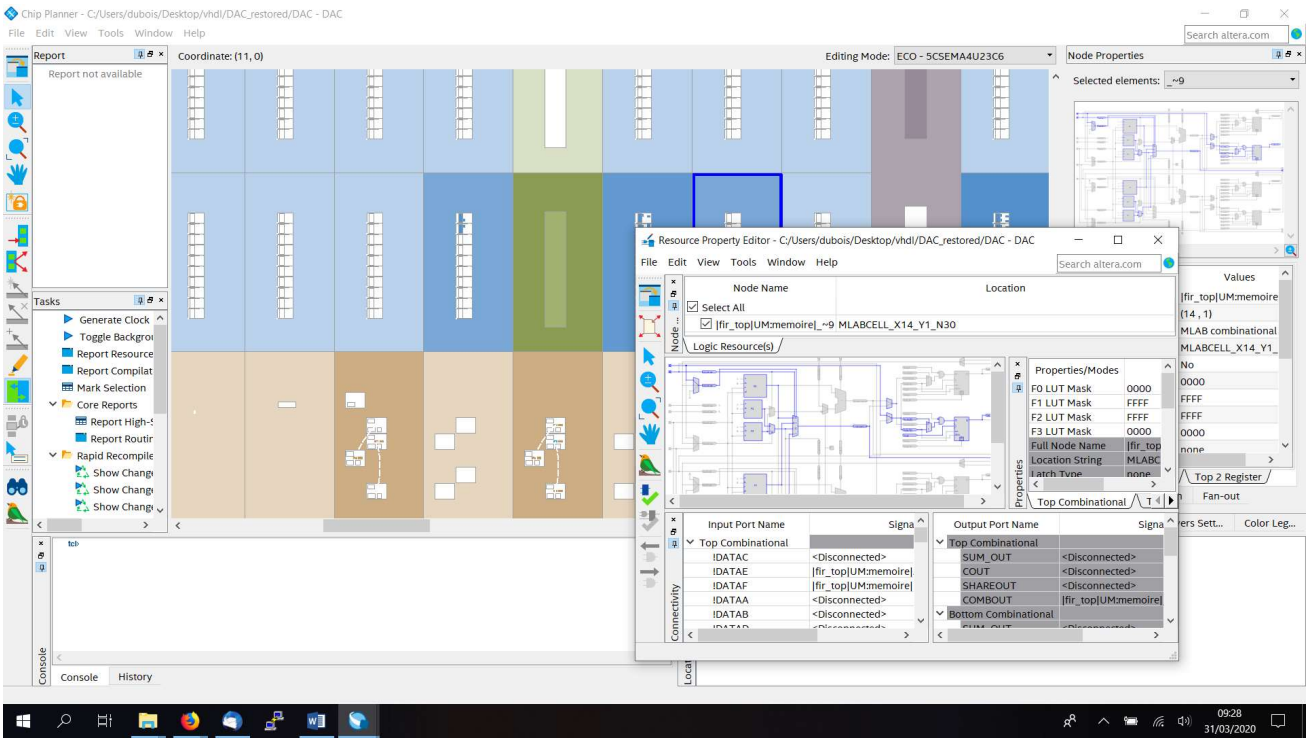
plusieurs derrière qui disparaitront à la correction de la première erreur. Prenez le temps de lire le message d'erreur qui, même s'il est en anglais, vous permet souvent de comprendre le problème. Pensez qu'une erreur sur une ligne peut provenir de la ligne précédente (exemple : oubli d'un point-virgule).
Si la compilation complète s'est bien passée, le rapport de compilation (fenêtre « compilation report) vous donne des informations très précises sur le design généré, notamment les ressources utilisées dans le FPGA par entité :



Vérifiez que le nombres de broches est correct, que le nombre de bascules (register) par entité est logique. Vérifiez que les bascules ont bien clk comme horloge, qu'il n'y a pas de latch (bascule sur niveau souvent générées à cause d'oubli dans des « if » ou des « case ».
Intéressez-vous à l'architecture du composant ciblé (voir sa documentation).
On voit par exemple ici qu'un DSP bloc a été utilisé (le compilateur a détecté qu'il y avait une multiplication accumulation) et que la mémoire de notre FIR a été placée dans des blocs mémoires embarqués dans le FPGA.

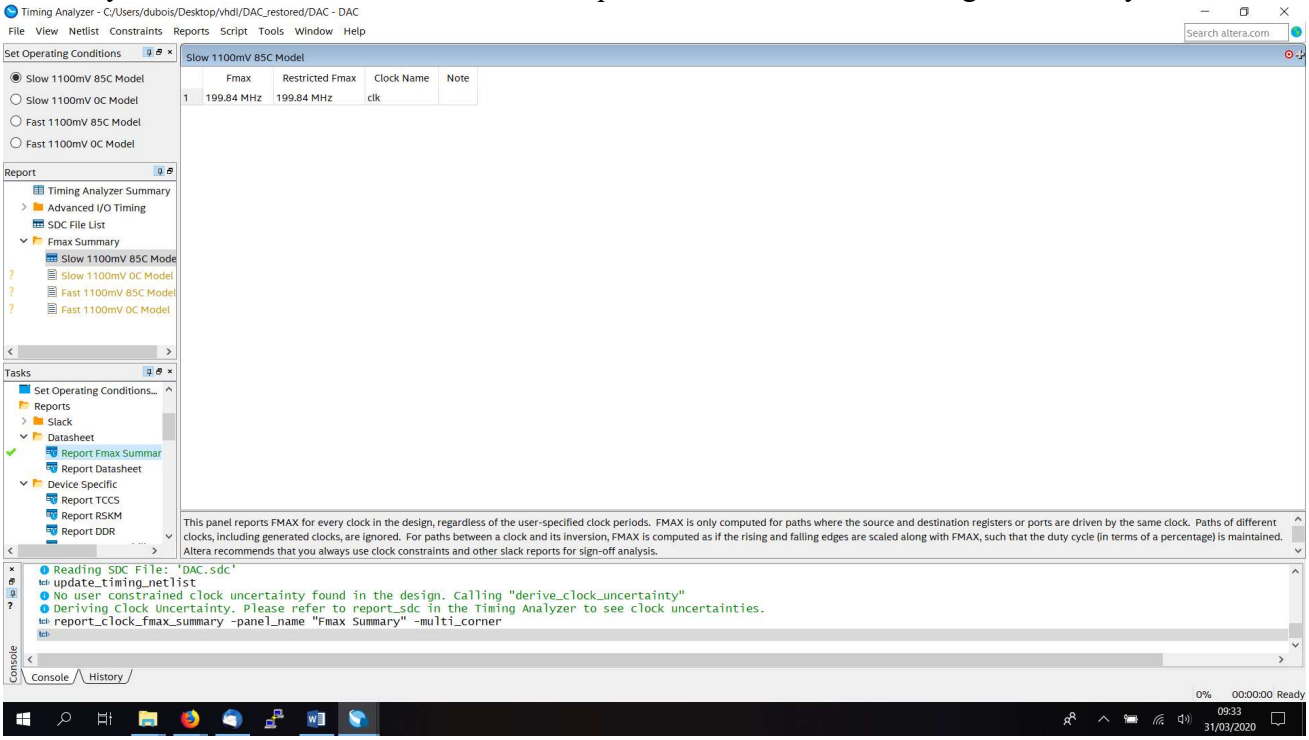
Un outil permet de voir graphiquement le placement-routage du FIR dans le FPGA : le Chip planner, accessible par le menu Tools. Lancez cet outil, reconnaissez sur le schéma le plan du FPGA ciblé. On voit en haut et à droite la partie SOC (Hard Processor System) que nous n'utilisons pas, on voit des colonnes colorées qui sont les colonnes de DSP blocs et de blocs mémoire réparties régulièrement dans le composant. Les cellules colorées plus foncées sont celles qui ont été utilisées pour placer le design. En zoomant sur un LAB et en cliquant sur une cellule, on peut voir jusqu'à l'intérieur d'une macro-cellule et comment elle a été configurée pour réaliser un petit morceau du FIR.

Utilisez quelques minutes cet outil pour bien comprendre le rôle de cette chaine Quartus qui est de réaliser dans une architecture FPGA précise, un système décrit ici en VHDL.
Vous pourrez regarder de même en projet comment a été implémenté le design de chaque version dans le FPGA.



3 Analyse de timing.

Le design ayant été placé et routé dans le FPGA, on peut maintenant connaitre la fréquence maximale à laquelle il peut fonctionner.
Lancez l'outil « TimeQuest Timing Analyzer » par le menu « Tools ». En cliquant sur « Report Fmax summary » vous obtiendrez la valeur de la fréquence max de la seule horloge de notre système Clk.

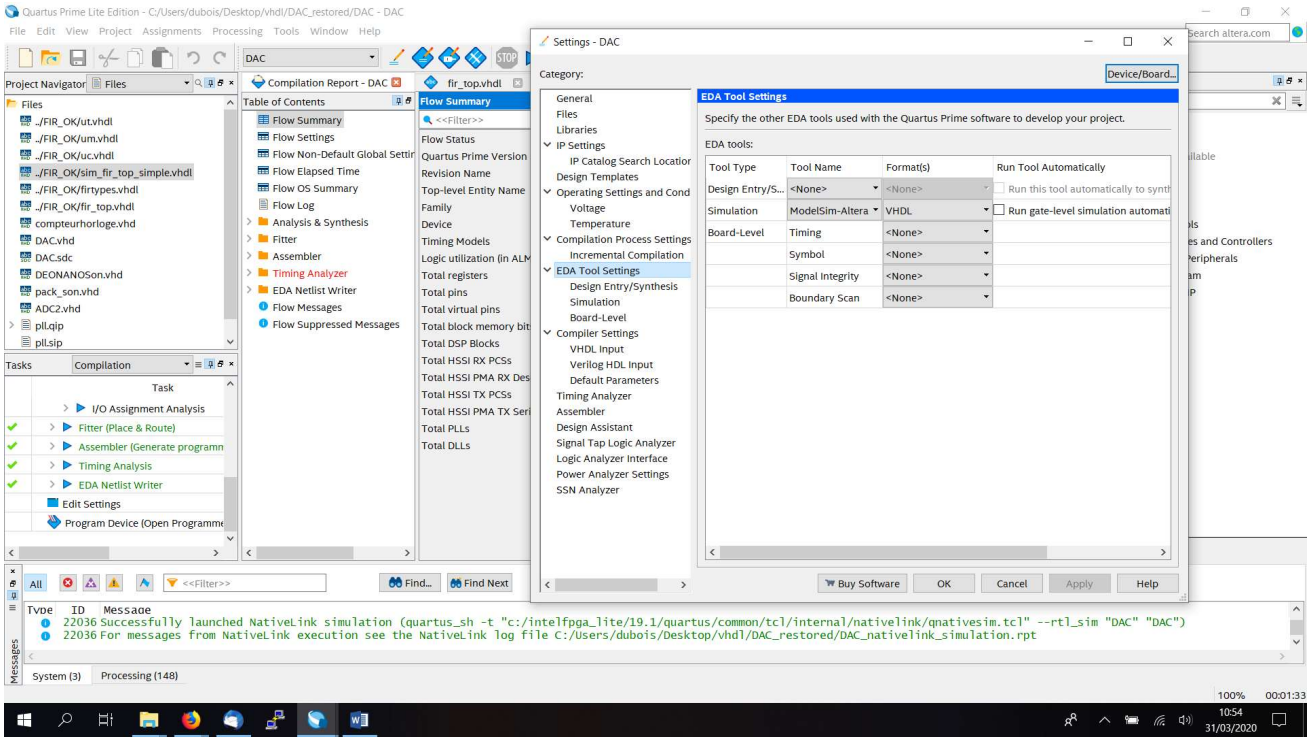


Vous pourrez en projet vérifier de la même façon que le design peut tourner à la fréquence de l'horloge disponible sur la carte (50 MHZ).

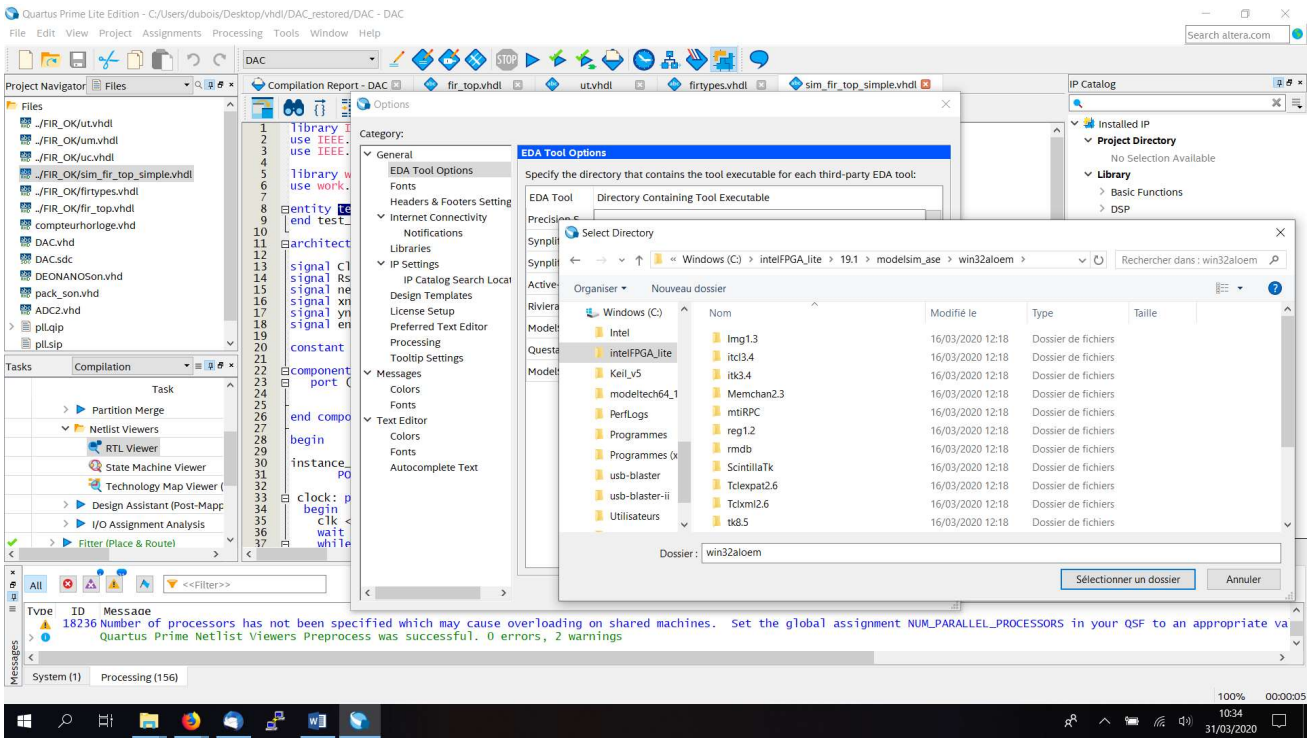
4 Simulation.

Les étapes précédentes garantissent que le VHDL est synthétisable et que le design tiendra dans le composant ciblé et pourra tourner à la fréquence visée.
Il faut également vérifier le fonctionnement avant d'aller sur la carte et cela grâce à des simulations.
L'outil de simulation sera Modelsim que vous avez déjà utilisé.
Notez bien qu'il ne sert à rien d'aller simuler avant d'avoir fait les étapes précédentes, car un code pourrait être correct en simulation, mais non synthétisable, VHDL étant un langage qui peut être utilisé pour autre chose que de la synthèse de circuits.

3 .1 Installation et rappels sur quelques points importants sur Modelsim.

- Modelsim peut être lancer seul ou à partir de Quartus. Les deux fonctionnements seront expliqués plus loin dans ce tutorial. On utilisera ici la version réduite Modelsim Altera qui a du être installée en même temps que Quartus. Vérifiez que vous avez bien le fichier Modelsim.exe sur votre machine.
- Sous Quartus, il faut indiquer que l'on utilise maintenant Modelsim Altera et pas la version complète de Modelsim.Il faut pour cela configurer deux écrans
    - L'écran EDA Tools Settings (accessible par Assignements/Settings)
- 

    - Et l'écran EDA Tools Options (accessible par le menu Tools/options)



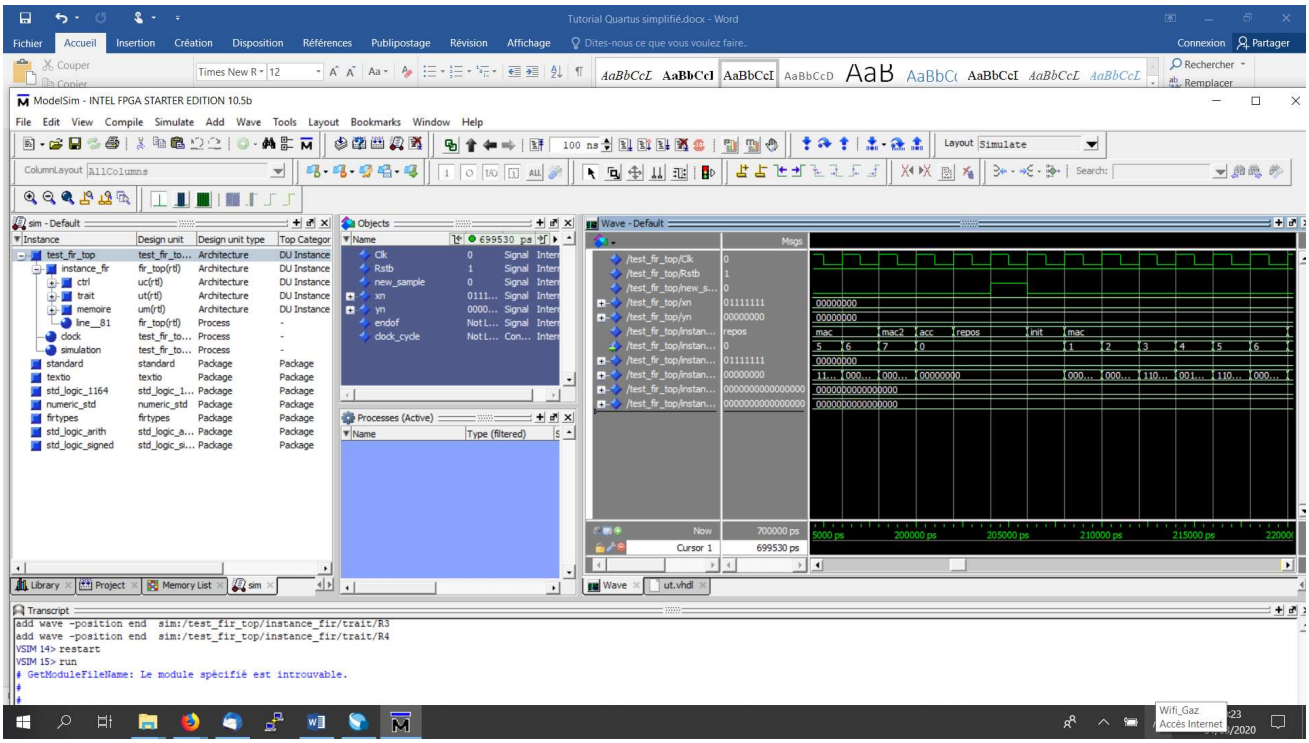
Modelsim sera alors « appelable à partir de Quartus.

Sous Modelsim :

- Le « prompt » dans la fenêtre « transcript » est « Modelsim> » si on est en mode commande et que la simulation n'est pas en cours ; il est « Vsim> » si on est en mode simulation. On a alors le temps où en est la simulation en bas à côté du « Now ».
- La visualisation des chronogrammes ne se fait que si la simulation est terminée et n'est pas en mode « run ». Elle se termine si tous les process sont terminés ou si vous appuyez sur l'icône « Stop » dans la barre d'outils.
- On peut créer un projet Modelsim qui indique les fichiers à compiler et permet de facilement recompiler par le menu « Compile/Compile all ». Ca fait gagner du temps et c'est donc conseillé.
- Si on veut pouvoir voir tous les signaux et ports de sa description, on a intérêt à lancer la simulation sans optimisation. Pour cela, lancez par le menu « Simulate/ Start simulation ». Sélectionnez le test bench et décochez « Enable optimisation » avant de lancer la simulation.
- Pensez à visualiser le maximum de signaux, ou tout du moins ceux qui vous permettent de vérifier le fonctionnement et de debbuger. Par exemple ici il est indispensable de visualiser la machine d'état, le compteur de boucle, etc. Pour cela, n'hésitez pas à descendre dans la hiérarchie du projet dans la fenêtre « Sim » , à sélectionner l'entité dans laquelle se trouvent les signaux qui vous intéressent et à les tirer sur la fenêtre Wave pour les visualiser. Il faut alors relancer la simulation pour voir ces nouveaux signaux : menu « simulate/restart » (cela remet le temps de simulation à 0).
- Si vous avez inséré pleins de signaux dans la fenêtre wave et que vous ne voulez pas les perdre mais que vous avez besoin de recompiler suite à une modification dans les sources VHDL, ne quittez pas le mode simulation, mais suspendez-le simplement par le menu « simulate/Break ». Compilez vos fichiers par le menu « compile », et relancez la simulation par le menu « simulate/restart ». Ca peut paraitre un détail, mais ça vous fera gagner beaucoup de temps quand vous corrigerez des erreurs.

3.1 Simulation avant synthèse.

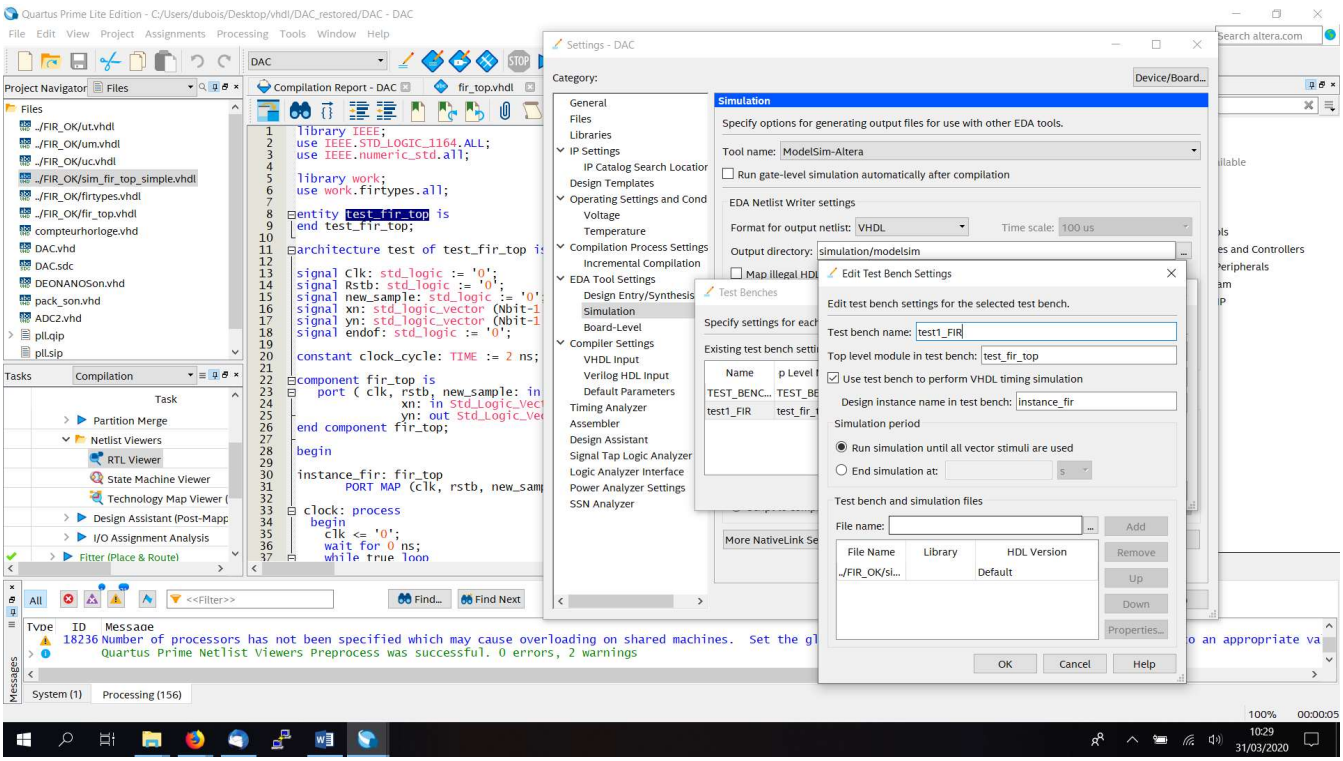
La première simulation va utiliser Modelsim tout seul , en compilant sous Modelsim . On a alors une simulation du VHDL source , assez rapide à obtenir et qui va permettre de debbuger beaucoup d'erreurs. Lancez Modelsim, créez un nouveau projet grâce au menu « File/New /Project ». Insérez les fichiers nécessaires, précisez le compile order, puis compilez par un « Compile/Compile all », puis lancez la simulation et insérez dans la fenêtre Wave tous les signaux intéressants pour le debbugage comme indiqué dans les rappels.
Voilà par exemple une simulation qui visualise la machine d'état, les registres de l'UT …



Vérifiez avec attention que le FIR fonctionne correctement en regardant à chaque front d'horloge si la machine d'état suit la bonne évolution et si le contenu des registres est correct, etc.

3.2 Simulation niveau RTL.

Si la simulation précédente est satisfaisante, il faut terminer par une simulation après synthèse en lançant Modelsim à partir de Quartus en niveau RTL. C'est alors le design RTL issu de la synthèse qui est simulé et plus le VHDL source. Cette simulation est la validation finale du design avant d'aller sur la carte. Pour lancer la simulation à partir de Quartus, il faut avoir pour renseigner quelques écrans sous Quartus pour qu'il puisse lancer Modelsim en lui indiquant ce qu'il faut simuler avec des noms qui font référence à ceux utilisés dans le fichier test bench vhdl.
Cela se fait par le menu « Assignments/Settings/Simulation » puis en renseignant les différents écrans comme indiqué sur la copie d'écran ci-dessous.



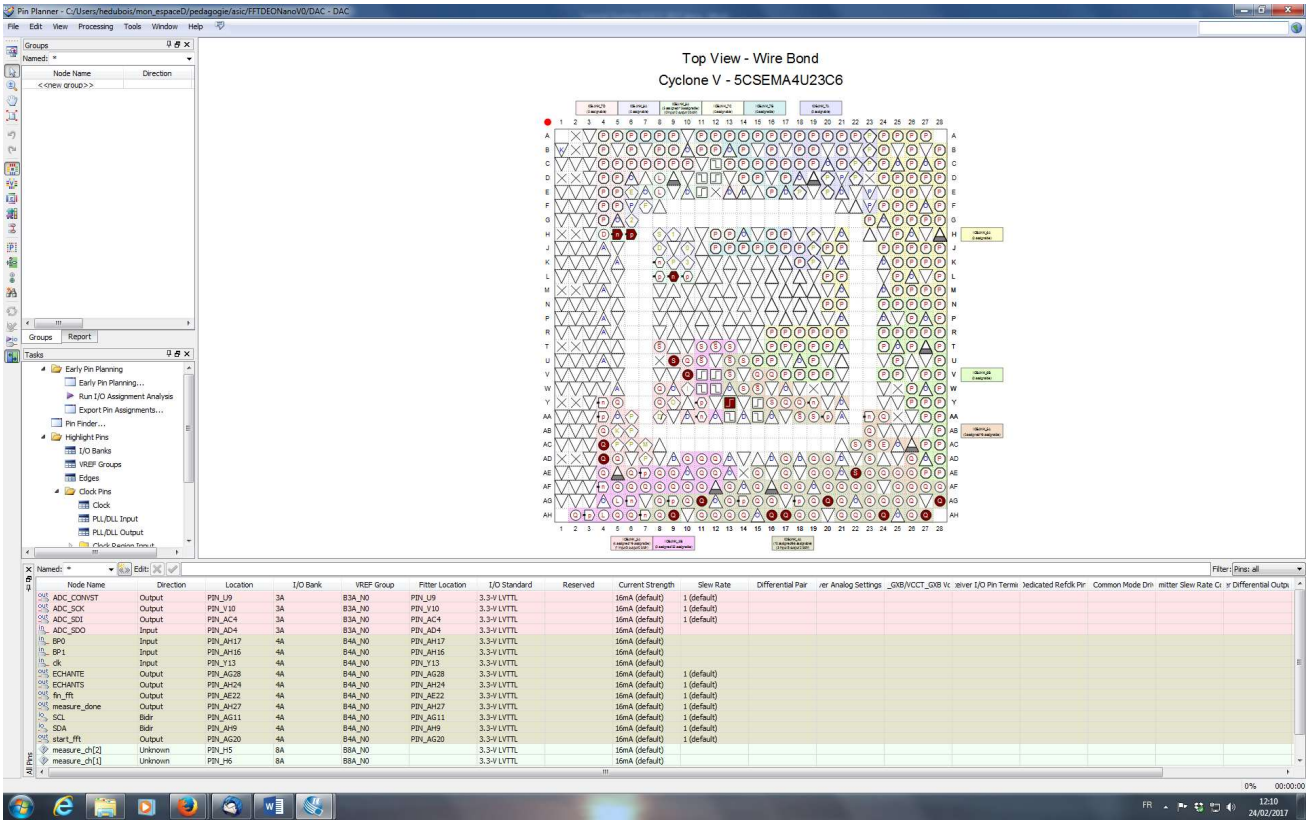
Faites la configuration de ces écrans en comprenant ce que vous faites (c'est-à-dire en voyant à quoi correspondent les noms indiqués dans « Top Level module in test bench » et « design instance name in test bench » par rapport au fichier de simulation), recompilez pour que les fichiers vers les « EDA Tools » soient correctement générés, puis lancez la simulation via le menu « Tools/Run simulation Tool/ RTL Simulation » de Quartus. Modelsim va être lancé via Quartus, puis compiler toutes les bibliothèques matérielles, ce qui est assez long …
Quand cela est terminé, vous êtes sous Modelsim en mode simulation et les ports d'entrée-sortie du design sont déjà visualisés. Vérifiez que la hiérarchie du projet est correspond à celle du design.
Si vous avez un message du type « component is not bound » dans la fenêtre transcript, ou que la hiérachy n'est pas bonne, revoyez attentivement la correspondance des noms entre le fichier test bench VHDL et l'écran précédent.

Si vous avez respecté dans votre conception les règles de la logique synchrone vues en cours et si vous mettez en simulation une fréquence de l'horloge CLK inférieure à la fréquence max trouvée par le timing Analyzer, cette simulation devrait être correcte du premier coup. Elle vous permet simplement de vérifier que la synthèse a donné une architecture RTL qui fonctionne.

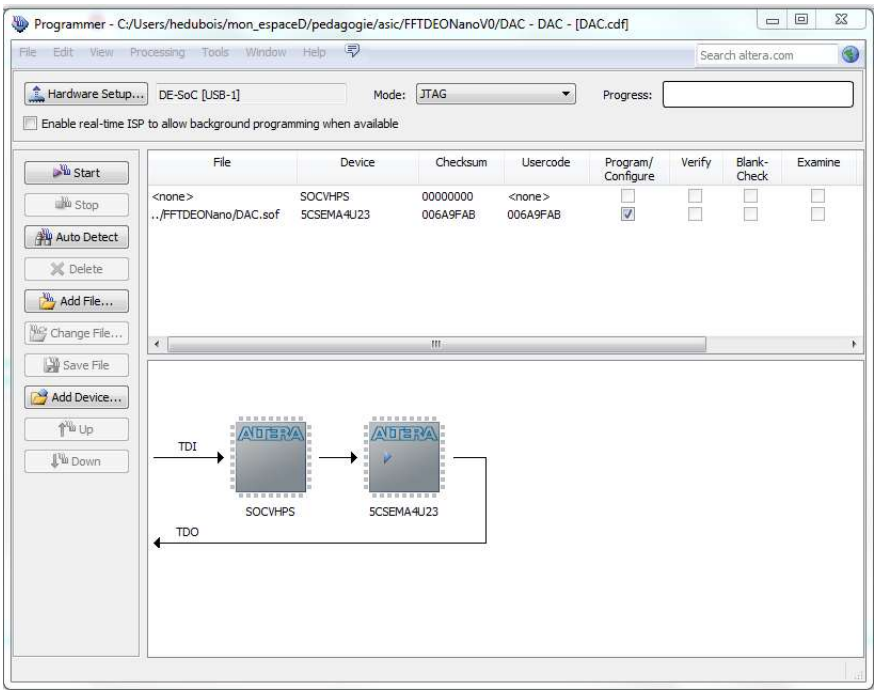
Les étapes précédemment montrées sont donc à faire quand vous allez écrire vos codes.
Vous avez là toutes les informations pour utiliser un seul projet Quartus et configurer les différents écrans pour travailler sur différentes top level entity, les compiler, les simuler avec différents tests benchs.
Attention de toujours bien vérifier la top level entity active et le test bench actif pour être sur d'analyser ce que vous croyez avoir été compilé ou simulé !

4 Programmation sur la carte.

Cette étape ne sera pas faite pour ce projet FIR, car elle ne présente pas d'intérêt, le projet n'intégrant pas l'interface avec les convertisseurs. Néanmoins voici comment on programme le FPGA à partir de Quartus.
Tout d'abord, il faut avoir pris soin de placer les entrées sorties du système sur des broches du FPGA compatible avec le PCB de la carte. Cela est fait par l'outil « Pin Planner » accessible par le menu « Assignement ». Pour le projet, le placement aura été fait en cohérence avec la carte DE0 NANO SOC. Un projet de départ avec les interfaces vers les convertisseurs et le placement des entrées et sorties vous sera fourni. Il faudra insérer vos codes dans ce projet.



Il faut ensuite utiliser l'outil « Programmer » accessible par le menu « Tools ». Après avoir relié et allumé la carte, cliquez sur « Hardware setup » et sélectionnez le mode de programmation par DEO SOC USB. Faîtes ensuite un « Add file » et sélectionnez le fichier .sof correspondant au projet. Ne laissez coché que le Program/Configure du composant 5CSEMA4U23 , puis cliquez sur « Start ». Vous pouvez voir l'avancement de la configuration du FPGA dans « Progress ».



Une fois cette configuration terminée, le FPGA bascule automatiquement en mode fonctionnement.

5 Archivage d'une version du projet.

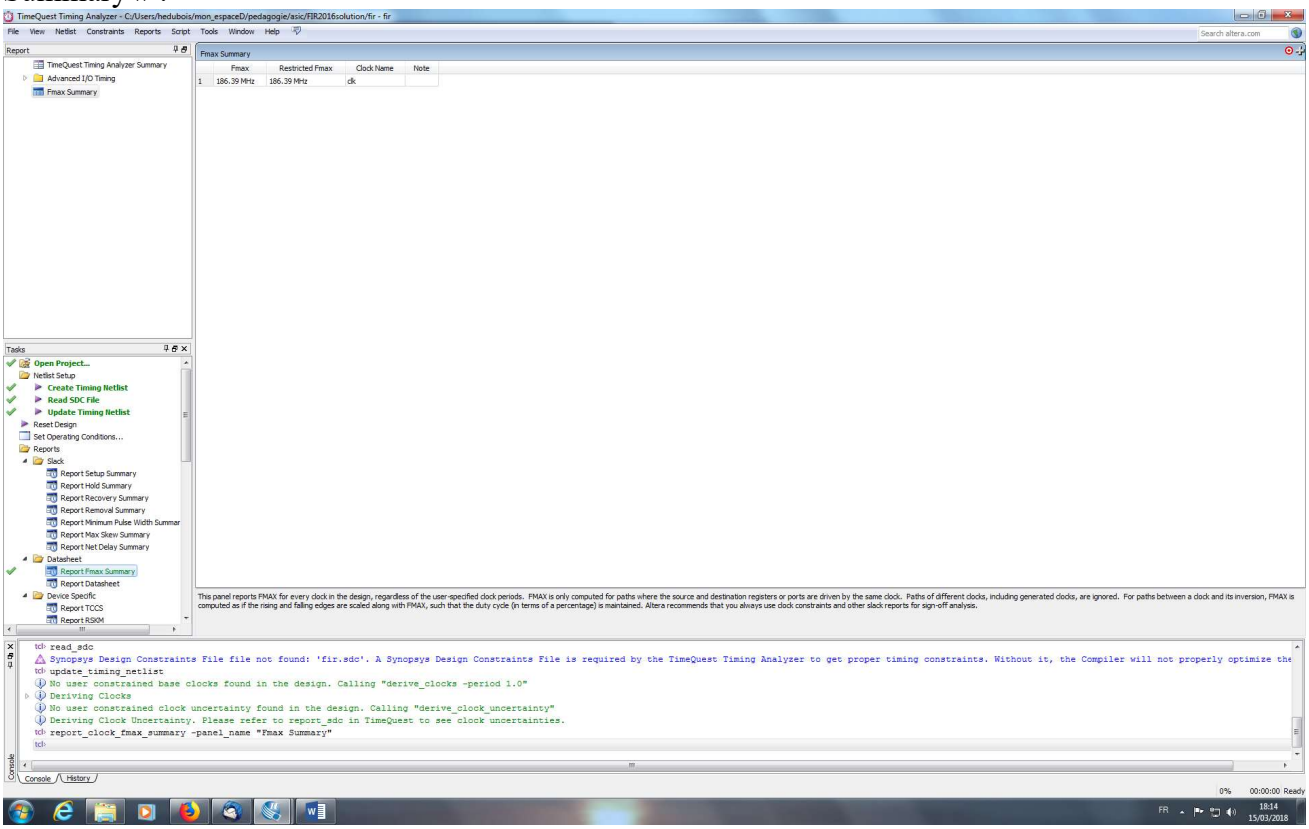
Une fois qu'une version est validée jusque sur la carte, il est conseillé d'en faire une archive.
Cela permettra en cas de problème sur la version suivante, de revenir à une version qui fonctionne.

Tout d'abord, vous pouvez simplement sauver le fichier .sof de la version qui fonctionne.
Ce fichier est celui qui sert à programmer le FPGA et il peut donc servir à reprogrammer le FPGA avec une version dont on sait qu'elle fonctionne pour vérifier que la carte marche toujours, ou si le client veut voir une version fonctionner alors que vous êtes en cours de développement d'une autre version.
Il est donc fortement conseillé de sauver le .sof de la V0 par exemple.

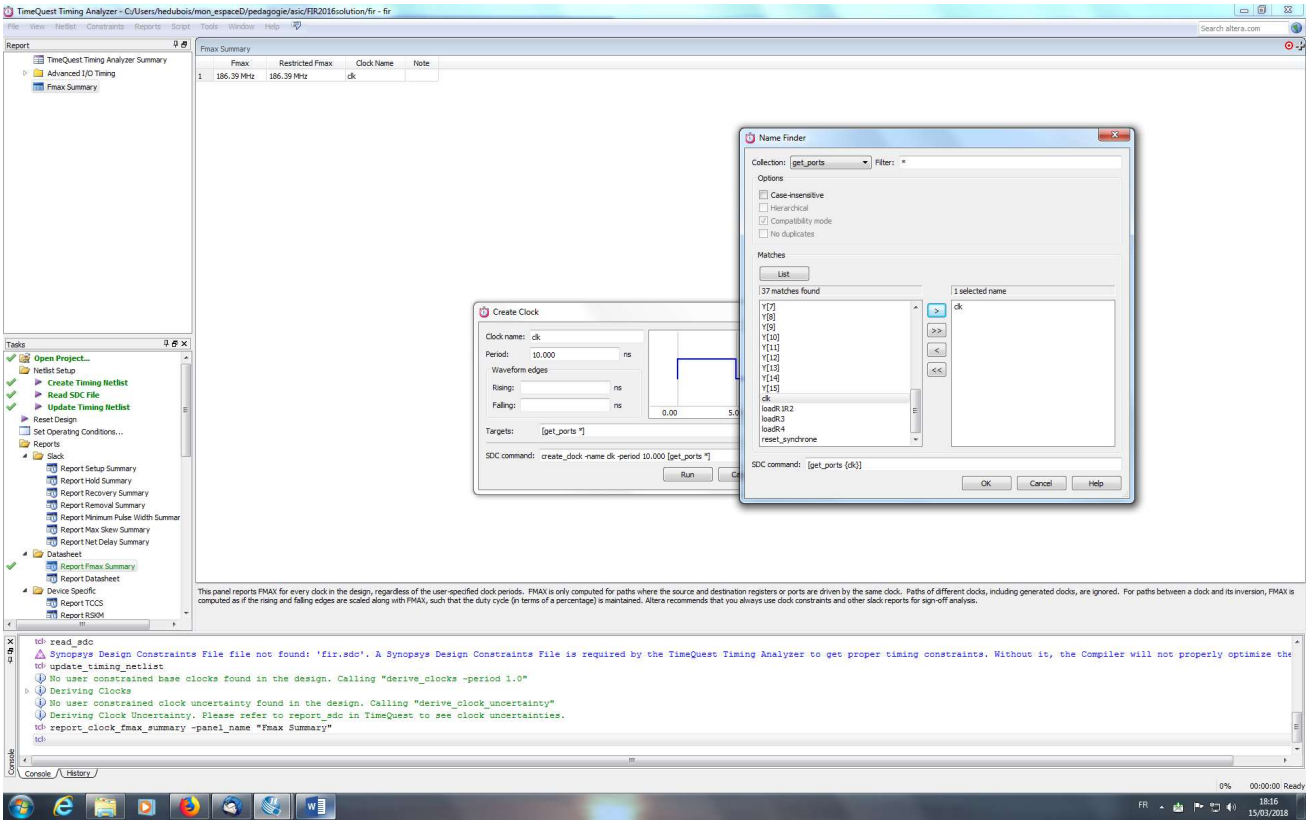
Ensuite l'ensemble d'un projet peut être très facilement sauvegardé par le menu Project/ Archive Project et régénéré par le menu Project/restore project.
Il est donc conseillé de créer une archive de chaque version validée avant de continuer sur une autre version.

Annexe : Obtention de la fréquence maximale et du chemin critique quand votre design sera OK en simulation.

L'outil TimeQuest Timing Analyzer permet d'analyser les performances temporelles du design.
Pour obtenir la fréquence maximale, il suffit d'ouvrir l'outil est de double cliquer sur « Report Fmax Summary » :

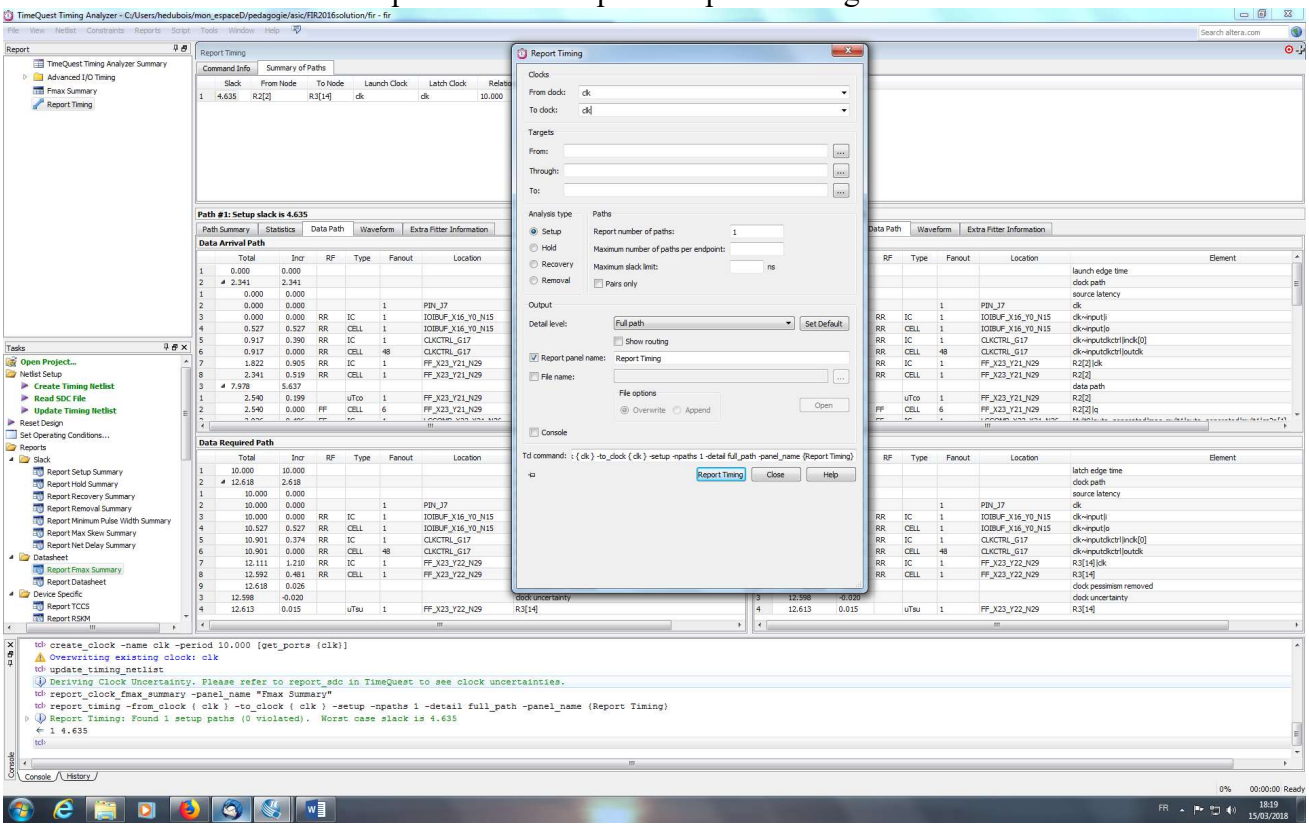


Pour obtenir le chemin critique, il faut tout d'abord faire l'équivalent du create clock sous Synopsys grace au menu Constraints/ Create Clock :



Le signal clk (qui doit être bien sur le nom du signal d'horloge dans votre design) est alors identifié comme l'horloge sur laquelle on va vérifier le slack sur les chemins entre les bascules.

Allez ensuite dans le menu Reports/Custom reports/Report Timing :



Vous obtiendrez le chemin critique, c'est-à-dire celui qui a le slack le plus faible.