Notice de prise en main du logiciel

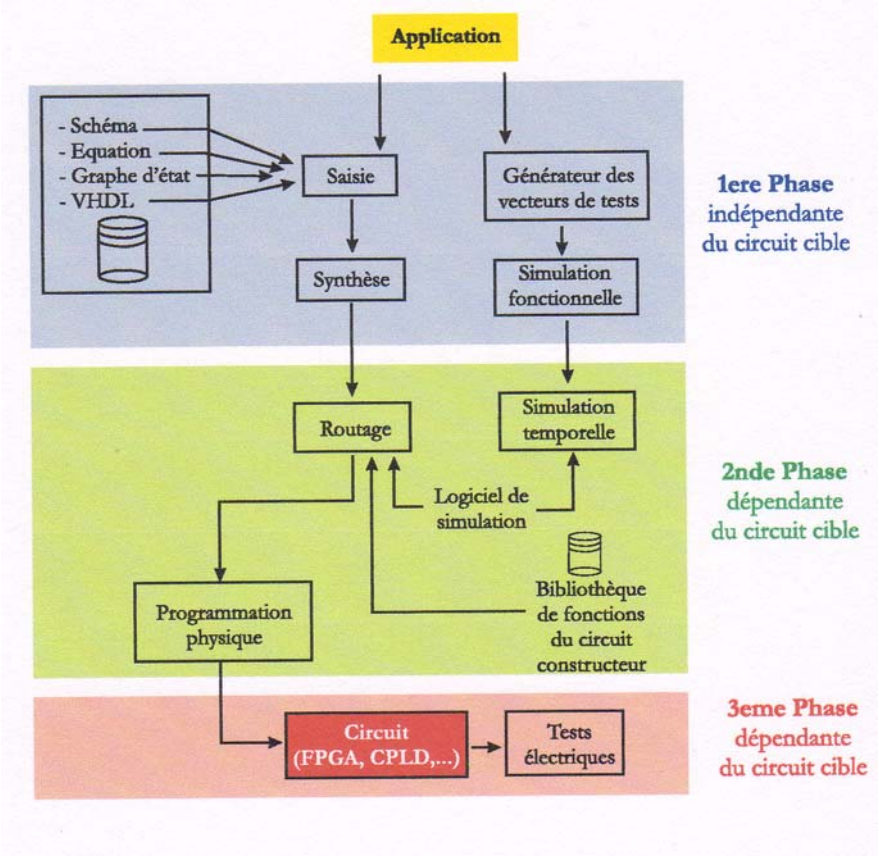
Quartus II

Table des matières

1. Présentation

Quartus est un logiciel développé par la société Altera, permettant la gestion complète d'un flot de conception CPLD ou FPGA. Ce logiciel permet de faire une saisie graphique ou une description HDL (VHDL ou verilog) d'architecture numérique, d'en réaliser une simulation, une synthèse et une implémentation sur cible reprogrammable.

Il comprend une suite de fonctions de conception au niveau système, permettant d'accéder à la large bibliothèque d'IP d'Altera et un moteur de placement-routage intégrant la technologie d'optimisation de la synthèse physique et des solutions de vérification. De manière générale, un flot de conception ayant pour but la configuration de composants programmables se déroulent de la manière suivante :



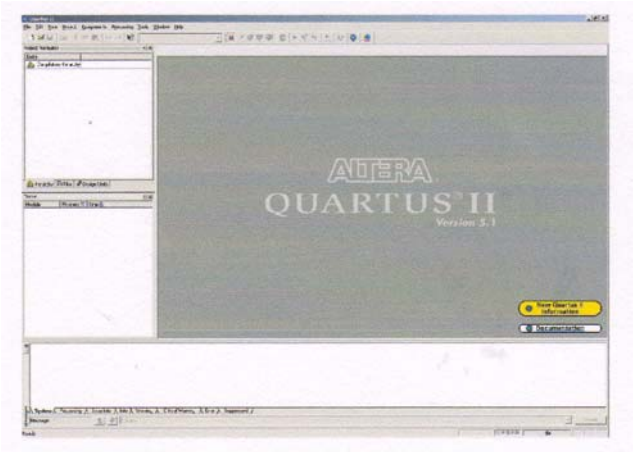
2. Création d'un projet

Quartus est un logiciel qui travaille sous forme de projets c'est à dire qu'il gère un design sous forme d'entités hiérarchiques. Un projet est l'ensemble des fichiers d'un design que ce soit des saisies graphiques, des fichiers VHDL ou bien encore des configurations de composants (affectation de pins par exemple).

Pour lancer le logiciel, on cliquera sur :

Démarrer —Programmes —>Altera —>Quartus II 5.1

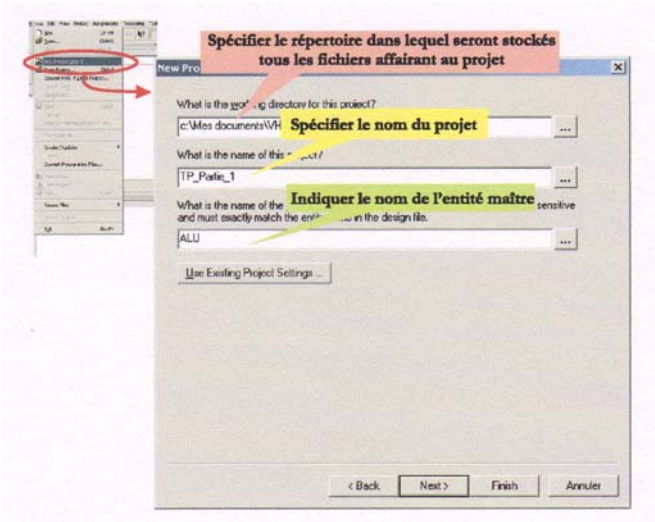
La fenêtre suivante s'ouvre :



Afin de créer un nouveau projet aller dans le menu :

File —>New Project Wizard

Puis se laisser guider. Une nouvelle fenêtre permettant de configurer le projet apparaît. Dans cette dernière trois champs sont à renseigner :



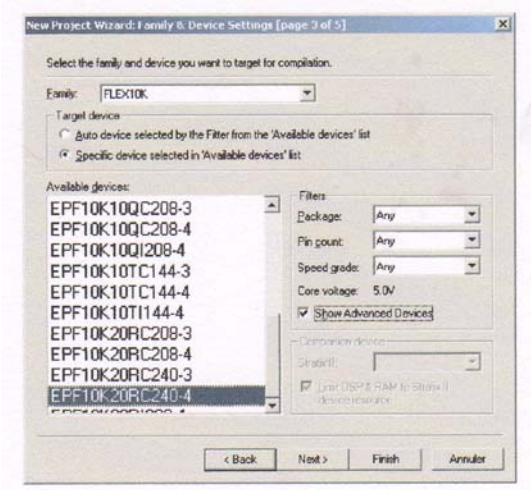
- L'emplacement du répertoire où seront stockés tous les fichiers. Il est conseillé de créer un répertoire propre afin d'assurer la sauvegarde des données d'une séance sur l'autre,
- Le nom du projet,
- Le nom de l'entité maître du projet. On appelle entité maître, le design qui correspond à la couche hiérarchique la plus haute. Par exemple s'il on conçoit un schéma nommé additionneur dans lequel il y aura plusieurs composants graphiques ou décrit en VHDL, alors additionneur sera l'entité maître. En d'autres termes, additionneur ne dépend pas d'autres fichiers, c'est le niveau le plus haut dans le design.

Cliquer sur **Next**, puis quand la fenêtre **Add Files** apparaît recliquer sur **Next**. Dans la fenêtre suivante intitulée **Family & Device Settings**, choisir le circuit logique programmable que l'on souhaite utiliser. Dans notre cas, nous choisirons un FPGA de la famille FLEX10K à savoir le EPF10K20RC240-4 qui se trouve être celui de la carte de TP.

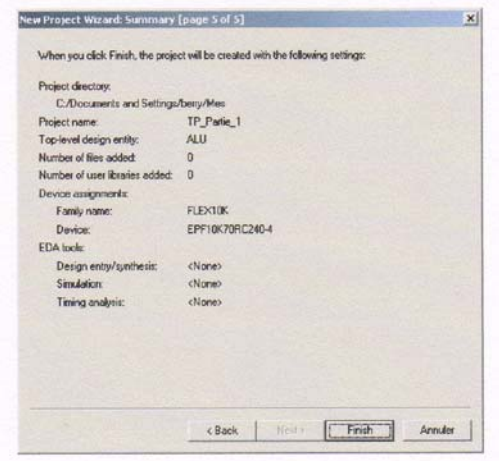
Bien que le nom du composant paraisse pour le moins exotique, cela nous renseigne sur :

- le fait que c'est un composant reprogrammable (EP) Composant Programmable et Effaçable (contrairement à la série Hard Copy HC),

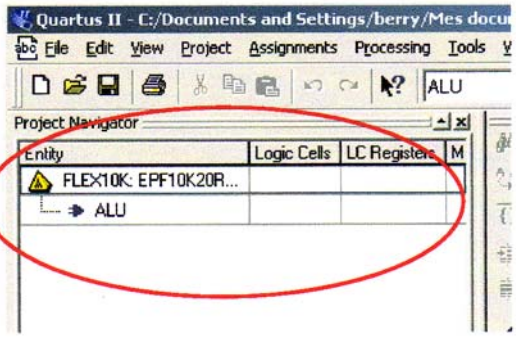
- le type de famille. Ici F10K correspond à la Famille FLEX10K
- le nombre de portes ici 20 000,
- le boîtier : RC correspond à un boîtier de type RQFP dit aussi boîtier "ailes de mouettes" (profils des broches),
- le nombre de broches : 240,
- la vitesse : -4. Plus le chiffre est petit (entre 2 et 4) et plus le circuit est rapide.



Quand la fenêtre **EDA Tool Settings** apparaît cliquer sur Next. Une fenêtre récapitulative apparaît :



Valider les choix par **Finish** ou bien faire **Back** pour des modifications éventuelles. Dans le navigateur de Projet, un onglet avec le type composant et l'entité maître apparaît :



3. Saisie d'un projet

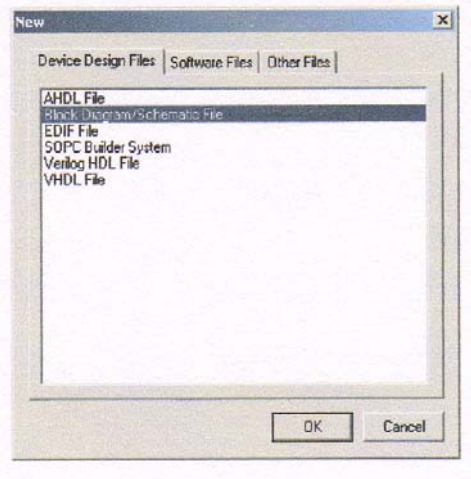
Cette étape permet de définir et configurer les différentes parties du projet. Quartus accepte plusieurs types de saisie à savoir :

- une saisie graphique en assemblant des symboles,
- une saisie textuelle à l'aide de différents langages (VHDL, Verilog, AHDL,...)

3.1. Saisie graphique

Pour saisir un projet en mode graphique, aller dans le menu : **File —>New**

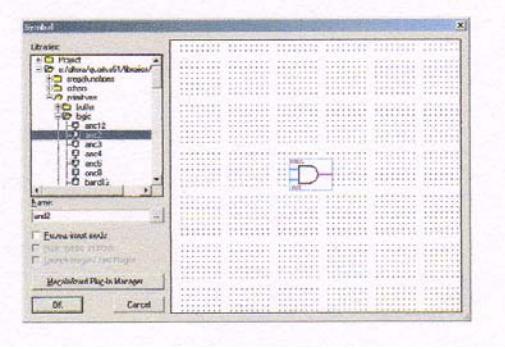
La fenêtre suivante apparaît :



Choisir **Block Diagram/Schematic File** et faire **OK**. Une feuille blanche se crée intitulée **Block1.bdf**. On prendra soin de sauver cette feuille sous le nom de l'entité maître (ALU exemple). C'est maintenant cette feuille de saisie graphique qui a la hiérarchie la plus haute dans le projet.

Il convient maintenant d'insérer des symboles dans notre feuille. Pour cela, nous pouvons soit choisir des composants de la librairie Altera soit en créer en les décrivant en VHDL.

L'insertion d'un symbole se fait en cliquant dans la feuille avec le bouton de droite et en allant dans **Insert —>Symbol**. La fenêtre suivante s'ouvre :



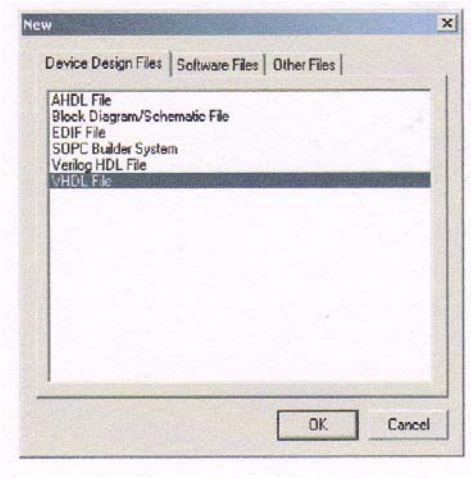
On remarquera sur l'onglet de gauche (Librairies), les bibliothèques propres au projet (personnel) et les bibliothèques natives d'Altera.

3.2. Saisie textuelle en VHDL

La saisie d'un composant VHDL se fait de la même manière que précédemment. Pour cela, aller dans le menu :

File —>New

La fenêtre suivante apparaît :



Choisir **VHDL File** et faire **OK**. Un petit éditeur de texte apparaît. Afin de fixer les idées, nous allons créer un petit composant réalisant un ET logique sur 4 bits par le programme suivant :

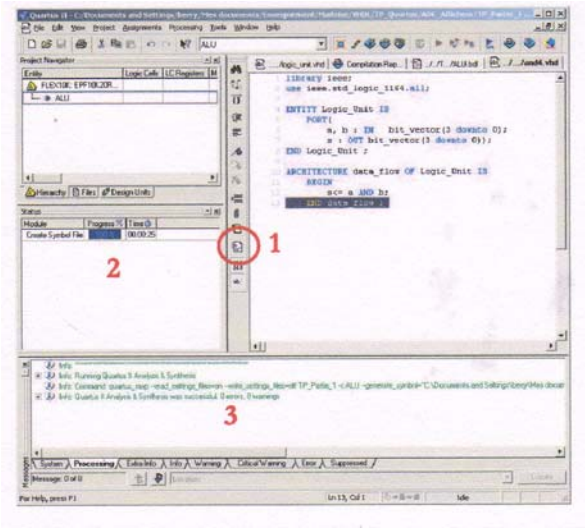
```
library ieee;
use ieee.std_logic_1164.all;

        ENTITY ET4 IS
                PORT(
                        a, b : IN     bit_vector(3 downto 0);
                        s : OUT bit_vector(3 downto 0));
        END ET4 ;

        ARCHITECTURE data_flow OF ET4 IS
                BEGIN
                        s<= a AND b;
                END data_flow ;
```

Une fois le code VHDL saisie, il convient de le sauver [1] (**File —>Save** sous le nom ET4 par exemple) puis d'en vérifier la syntaxe. Pour cela, on cliquera sur l'icône entouré noté 1 et l'on suivra l'évolution de la vérification en 2 et les différentes informations s'afficheront en 3.

[1] Il est important de sauver le fichier sous le même nom que l'entité. Bien que cela ne soit pas indispensable comme sous Maxplus, cela évite des intersections d'entité entre fichiers. Dans notre cas, l'entité s'appelle ET4 donc nous sauverons le fichier sous ET4.vhdl. On prendra aussi garde à ne pas appeler une entité du même nom qu'un composant natif Altera comme par exemple AND4.



Une fois que le fichier est OK, on peut alors créer un symbole graphique qui nous permettra d'insérer le composant dans la feuille graphique initiale (ALU). Pour cela, aller dans

File —»Create/Update —>Create Symbol File from Current File

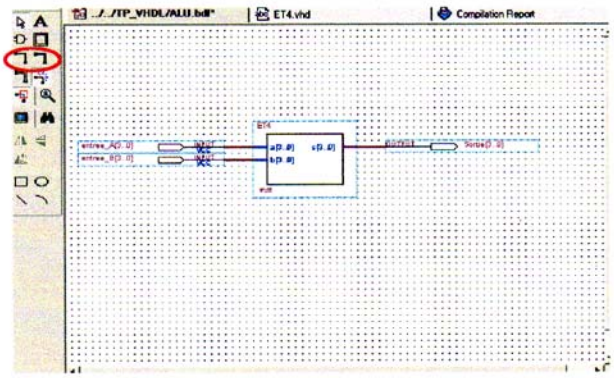
Le symbole correspondant à notre composant VHDL est maintenant créé. Nous pouvons l'instancier dans la feuille graphique ALU en l'insérant comme cela est expliqué précédemment. Normalement, le composant ET4 doit se trouver sous l'onglet Projet.

Afin d'en faire une simulation, on lui adjoindra des composants d'entrées/sorties (IO) permettant de repérer les signaux. Ces derniers se trouvent dans la librairie :

Altera —>Primitives —>Pin

La liaison entre les composants se fait à l'aide de l'icône entouré en rouge dans la fenêtre ci-dessous. On notera la différence entre les équipotentielles simples (trait fin) et les bus (trait large).

On doit se trouver avec un schéma de ce type :



Il est possible de renommer les pins en double-cliquant dessus. Une fenêtre de propriétés apparaît et il suffit de se laisser guider.

4. Compilation

Cette étape consiste maintenant à compiler le schéma précédemment réalisé.

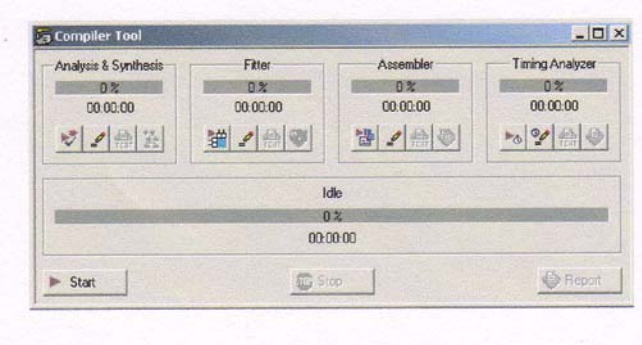
Durant la compilation, Quartus va réaliser 4 étapes :

- La transformation des descriptions graphiques et textuelles en une structure électronique à base de portes et de registres : C'est la synthèse logique.
- L'étape de Fitting (ajustement) consiste à voir comment les différentes portes et registres (produit par la synthèse logique) peuvent être placés en fonction des ressources matérielles du circuit cible (FLEX 10K20) : C'est la synthèse physique.
- L'assemblage consiste à produire les fichiers permettant la programmation du circuit. Ce sont des fichiers au format Programmer Object Files (.pof), SRAM Object Files (.sof), Hexadecimal (Intel-Format) Output Files (.hexout), Tabular Text Files (.ttf), and Raw Binary Files (.rbf). Dans notre cas, nous utiliserons toujours le format SOF pour les FPGA et le format POF pour les CPLD.
- L'analyse temporelle permet d'évaluer les temps de propagation entre portes et le long des chemins choisis lors de l'étape de « fitting ».....

Pour lancer la compilation, cliquer sur :

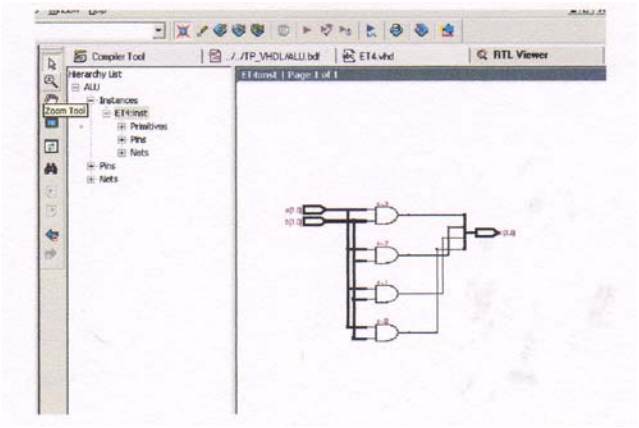
Processing —>Compiler Tool.

La fenêtre suivante apparait. On cliquera sur Start pour la compilation.



Normalement il ne doit pas y avoir de warning ou d'erreur. Si ce n'est pas le cas vérifiez dans la zone Processing (en bas où s'affichent les messages) la source du problème. En cliquant sur **Report,** on trouve une multitude d'information entre autres le pourcentage d'occupation du schéma dans le circuit programmable, les temps de propagation, les fréquences maximums d'utilisation etc…

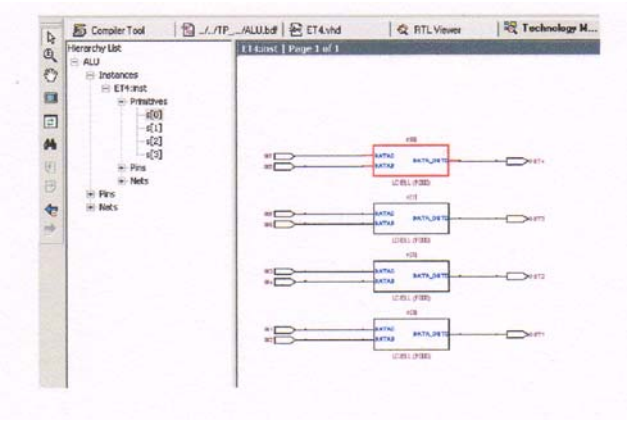
Afin d'illustrer cette partie, on pourra par exemple lancer la commande **Tools —>RTL Viewer** qui permet de voir comment le schéma (ALU) contenant le code VHDL a été transformé en portes et bascules. Cela permet de voir comment la synthèse logique s'est déroulée. Dans notre cas, nous retrouverons bien les 4 portes AND.



De même, nous pouvons visualiser la synthèse physique en utilisant la commande :

Tools —>Technology Map Viewer

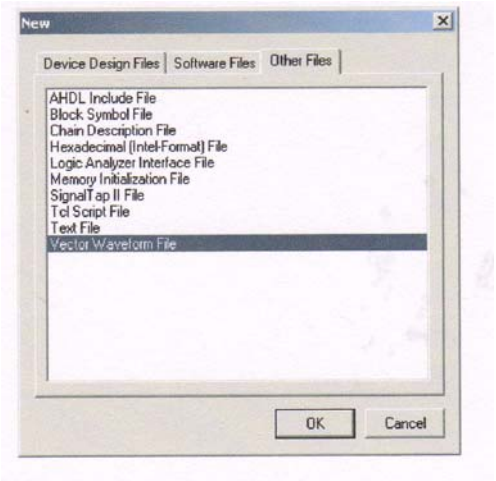
On retrouve les 4 instances placées dans le circuit et repérées par leurs références.



5. Simulation

Afin de simuler le design réalisé, il convient de lui injecter des stimuli. Lorsque ces stimuli sont générés à partir d'un fichier on dit que l'on utilise un fichier de Bench.

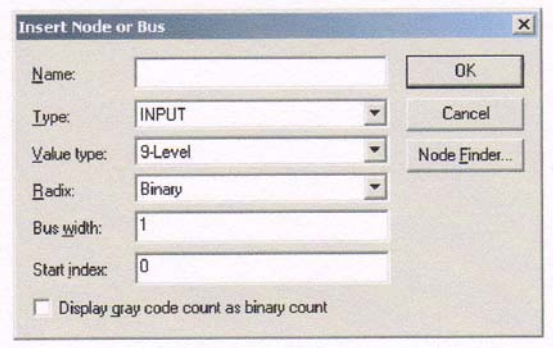
Dans le cas présent, nous allons simuler notre schéma en générant les stimuli à partir du Wave *Editor.* Pour cela, faites **File —>New**, aller dans l'onglet **Other Files** et sélectionnez **Vector Waveform File** (Fichier de simulation).



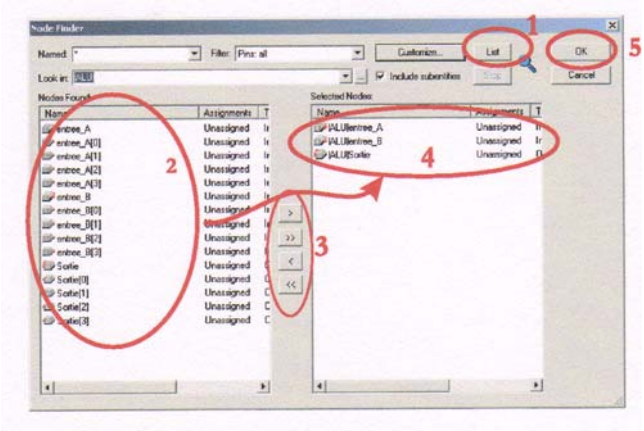
Faire une sauvegarde du fichier par **File —>Save** en lui donnant un nom compréhensible (par exemple Simul_ALU). Le fichier sera du type .vwf.

Pour changer la durée de simulation, on fera **Edit —>End Time** et pour réaliser des zooms ou voir toute la simulation, il est possible de cliquer dans la fenêtre avec le bouton de droite et de choisir **Zoom** dans le menu.

Il ne reste plus qu'à insérer les différents signaux de simulation c'est à dire les signaux d'entrée et les sorties. Pour cela, on fera **Edit —>Insert Node or Bus**. Dans la fenêtre



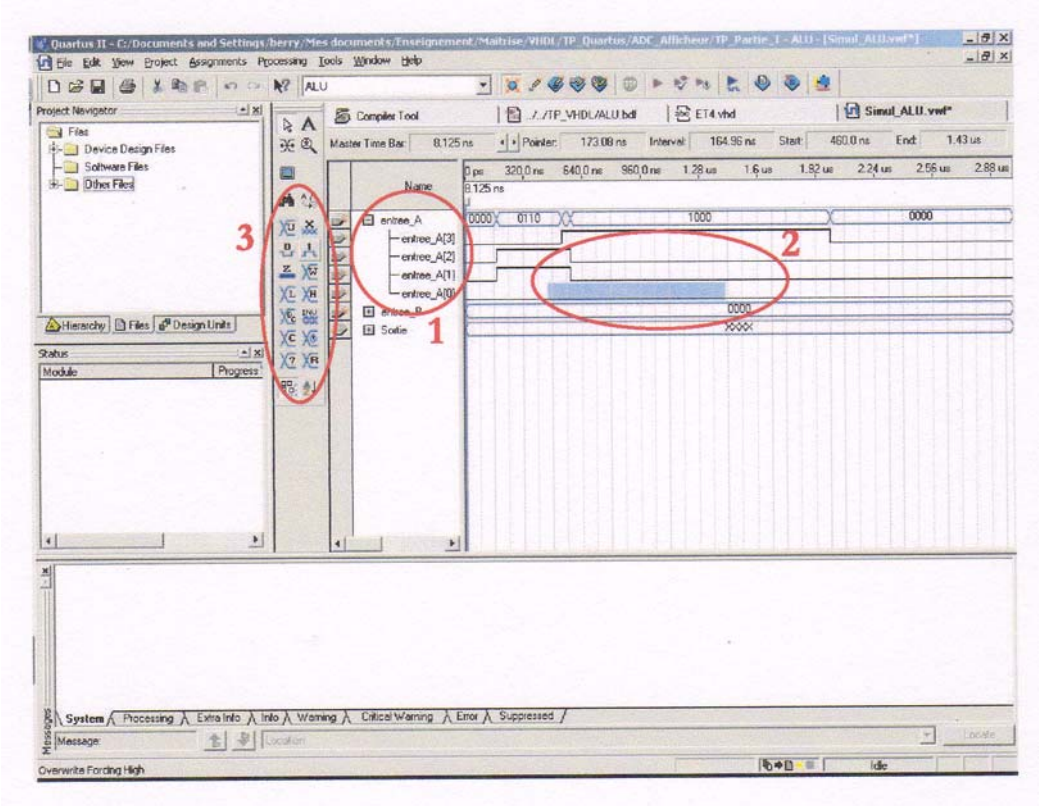
Cliquer sur **Node Finder**, ce qui permet de lancer le navigateur de signaux



Dans le **Node Finder**, cela se déroule en 5 étapes :

1. Cliquer sur list afin de faire apparaître les signaux du design,
2. Sélectionner les signaux voulus,
3. Cliquer sur la flèche correspondante
4. Vérifier que tous les signaux que vous voulez visualiser sont dans Selected Nodes
5. Valider par OK Cliquer sur OK dans la fenêtre Insert Node or Bus

Les signaux apparaissent maintenant dans le visualiseur de signal. Si des bus ont été sélectionnés, il est possible de les "déplier" en bit à bit en cliquant sur le + (Zone 1).



Afin de donner des valeurs de stimuli, on sélectionne à la souris une partie du signal (2) et avec le menu (3) on lui attribue une valeur. On notera la possibilité d'insérer automatiquement une horloge (clock) ou un compteur.

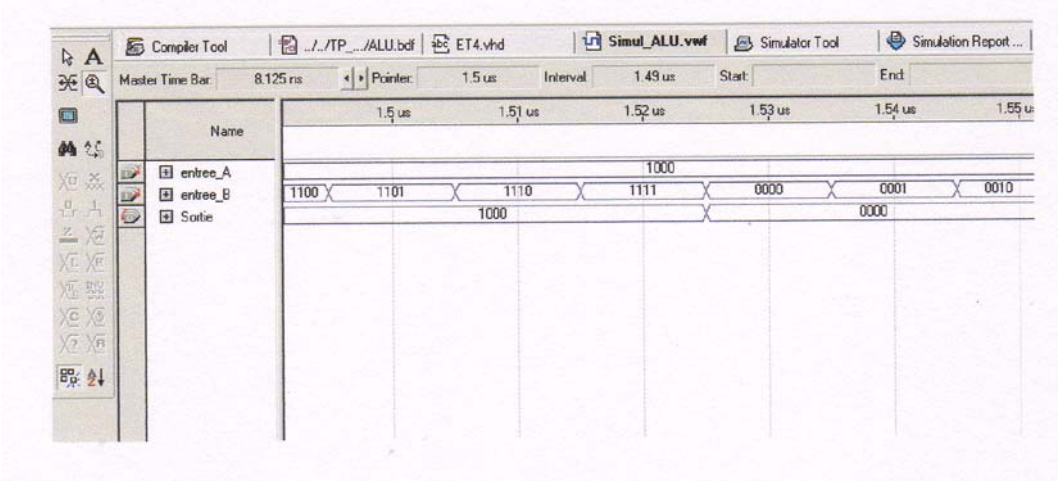
En dernier lieu, il ne reste plus qu'à lancer la simulation par la commande :

Tools →Simulator Tool.

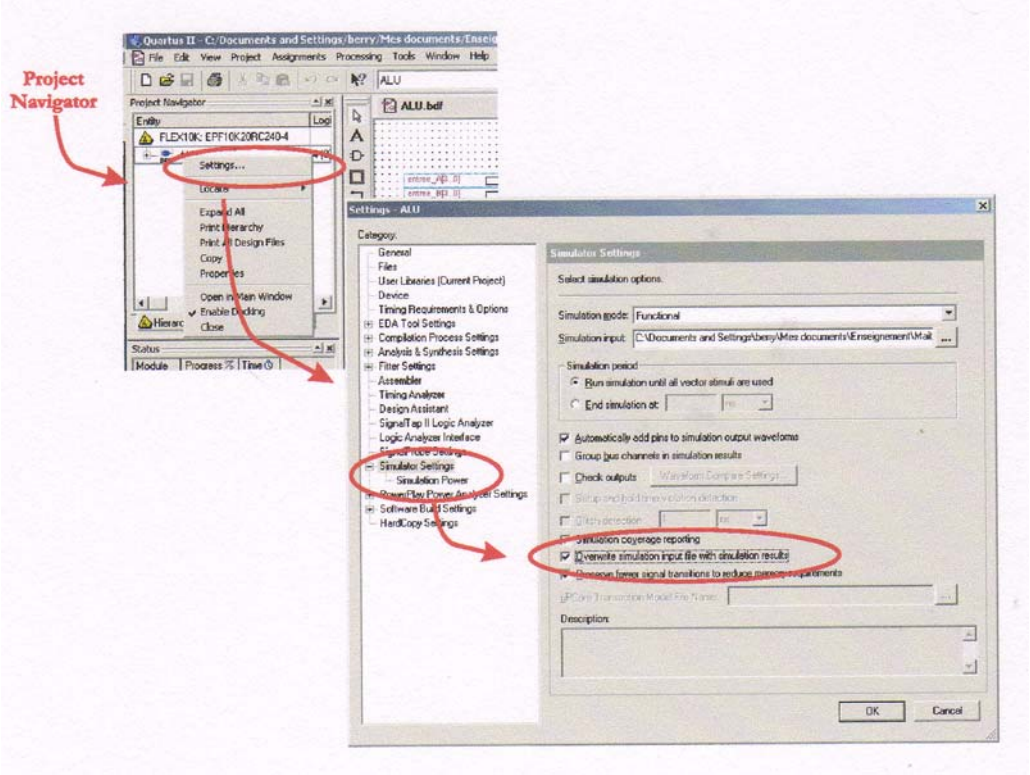
Dans cette fenêtre, il faut tout d'abord spécifier le fichier de simulation (par exemple Simul_ALU.wvf) dans la partie 1.

La zone 2 permet de choisir si l'on veut prendre en compte les délais (timing) ou non (functional). Dans ce dernier cas, il faut générer un modèle fonctionnel en cliquant sur le bouton 3.

Le bouton **Start**(4) permet de lancer la simulation et il est possible de voir le résultat en cliquant sur **Report**(5).



Afin de permettre le rafraîchissement automatique du visualiseur de signal, il suffit de modifier une option en cliquant avec le bouton de droite sur l'entité maître (ALU) dans le Project Navigator, puis en choisissant le menu **Settings**. Dans la nouvelle fenêtre aller dans l'onglet **Simulator Settings** et cocher la case **Overwrite simulation input file with simulation results**



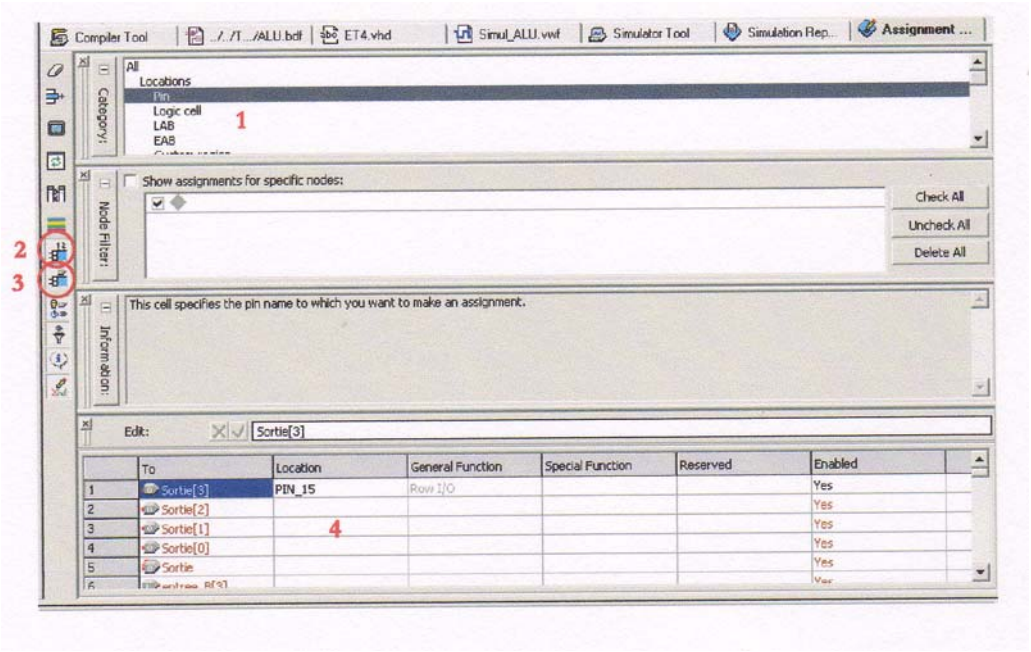
6. Programmation d'un circuit

C'est l'étape ultime. Pour cela, il faut assigner les pins d'entrée/sortie du design aux broches du circuit physique. Si rien n'est spécifié, Quartus affectera comme bon lui semble des pins (A éviter!).

6.1. Affectation des pins

Afin de choisir quelle broche physique du circuit doit être connectée, lancer l'outil d'assignement de pins par :

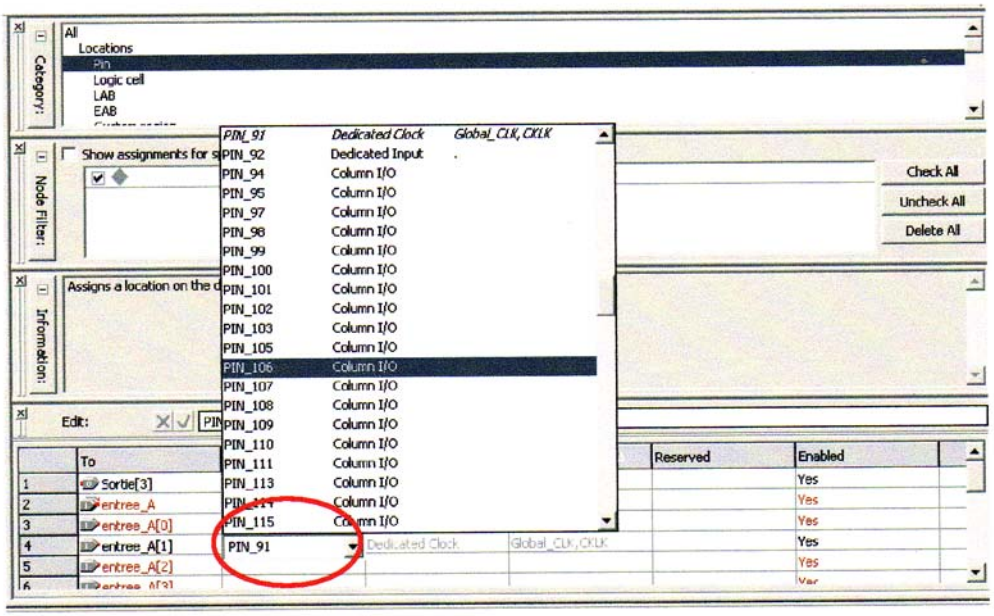
Assignements →Pins



Dans la fenêtre correspondante, il est possible de choisir différents types d'assignement.

Dans notre cas, nous sélectionnerons dans la partie 1 l'onglet Pin. Dans le menu de gauche, prendre garde à dévalider le bouton (2) permettant de faire afficher toutes les pins et bien valider le bouton (3) ne permettant l'affichage que des pins utilisées. De cette manière dans la partie 4 ne doivent s'afficher que les pins utilisées dans le design.

Afin de réaliser l'affectation, on double clique dans la colonne **Location** de la partie 4 au niveau de la pin voulue de manière à faire apparaître un menu déroulant où sont répertoriées les broches disponibles du circuit.



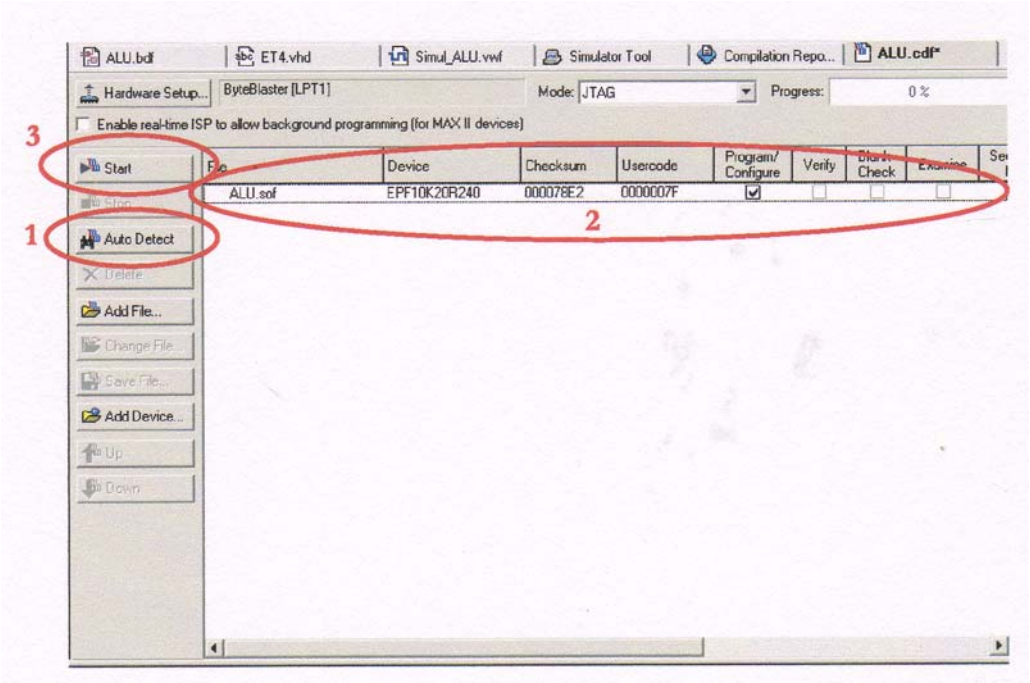
La liste des broches utilisables pour le FPGA et sortant sur les connecteurs est données en annexe. On notera que certaines broches sont réservées pour des horloges, des resets, des CS (Chip Select),... ou sont directement câbles sur des périphériques tels que les afficheurs, les boutons poussoirs,...

Dans le cas du design de la porte NAND, on peut par exemple câbler sur chacun des poussoirs un bit de l'entrée A et un bit de l'entrée B, puis câbler la sortie sur une led de l'afficheur.

On notera que l'état bas est à 5V et l'état haut est à 0V sans mettre des inverseurs aux entrées et sorites.

6.2. Programmation du circuit

La programmation du circuit se fait via le protocole JTAG (Joint Test Action Group). Pour cela vérifier que les connections entre le PC (port parallèle) et la carte via le module **ByteBlaster** sont opérationnelles. Si tout est bon et que la carte Altera est sous tension, lancer le programmateur par **Tools →Programmer**, puis cliquer sur le bouton **Autodetect** (1). Si le PC ne détecte pas la carte, une erreur doit apparaître du type Unable to scan device Chain. Hardware is not connected.



Vérifier dans la partie 2 que le fichier .sof est bien là et que la case **Program/Configure** est cochée, puis cliquer sur **Start**.