

ispLEVER Classic 1.2, Active-HDL 8.1LWE et ispVM System : tutorial

Frédéric Senny – 20/10/2008

ispLEVER Classic 1.2

Ce logiciel est fourni gratuitement par Lattice via www.latticesemi.com. Afin de créer la licence (gratuite), Lattice demande de lui fournir l'adresse MAC (ou NIC address) du PC.

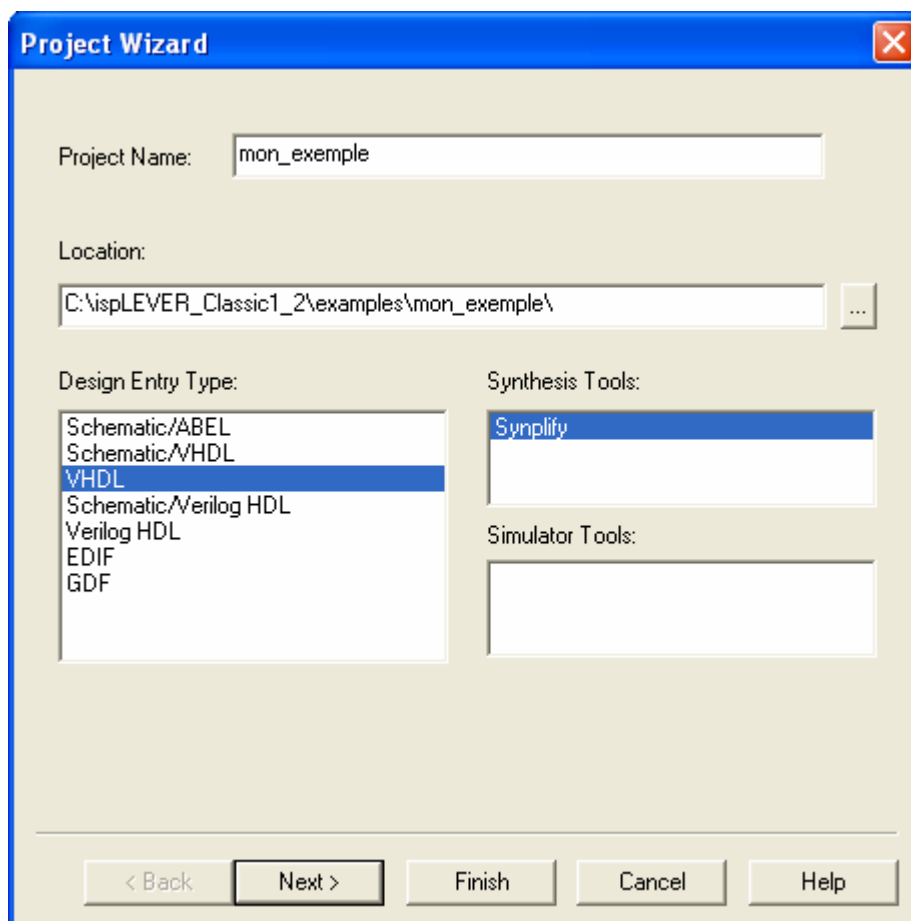
ispLEVER Classic est une suite d'outils qui permet la création du fichier (JEDEC par exemple), utilisé pour la programmation à proprement parlé du composant. Les composants supportés par cette suite sont bien évidemment ceux de Lattice, et plus particulièrement les composants éprouvés (les grands classiques en somme).

Avec ispLEVER, vous

- créez un projet contenant la description VHDL du projet,
- compilez le VHDL et pourrez le corriger,
- créez le fichier JEDEC nécessaire à la programmation de la (isp)GAL22V10C grâce à l'outil du dataman PRO+ ou ispVM System.

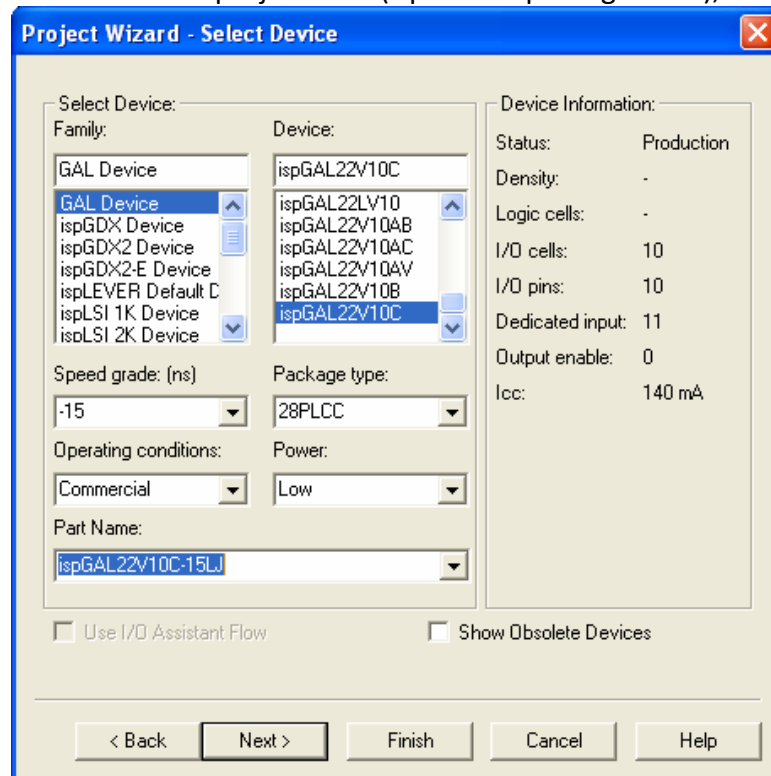
1. Création du projet

File → *New Project ...* → donnez un nom au projet (mon_exemple), son chemin (C:\ispLEVER_Classic1_2\example\mon_exemple\), son type (VHDL) et choisissez l'outil de synthèse (synplify) → *Next*



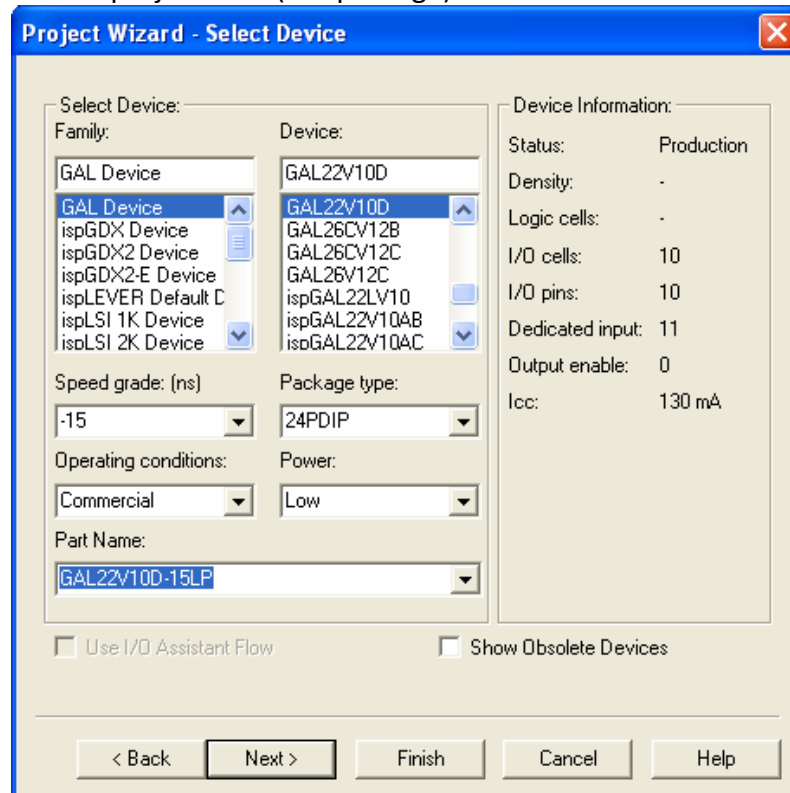
2. Choix du composant

Pour le labo ou projet VHDL (ispGAL en package PLCC),



The 'Project Wizard - Select Device' dialog box is shown. The 'Family' list on the left includes 'GAL Device', 'ispGDX Device', 'ispGDX2 Device', 'ispGDX2-E Device', 'ispLEVER Default C', 'ispLSI 1K Device', and 'ispLSI 2K Device'. The 'Device' list on the right includes 'ispGAL22V10C', 'ispGAL22LV10', 'ispGAL22V10AB', 'ispGAL22V10AC', 'ispGAL22V10AV', 'ispGAL22V10B', and 'ispGAL22V10C'. The 'Speed grade: (ns)' is set to '-15', 'Package type' is '28PLCC', 'Operating conditions' is 'Commercial', and 'Power' is 'Low'. The 'Part Name' field shows 'ispGAL22V10C-15LJ'. The 'Device Information' section on the right shows: Status: Production, Density: -, Logic cells: -, I/O cells: 10, I/O pins: 10, Dedicated input: 11, Output enable: 0, and Icc: 140 mA. At the bottom, there are checkboxes for 'Use I/O Assistant Flow' and 'Show Obsolete Devices', and buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Pour le projet VHDL (DIP package)

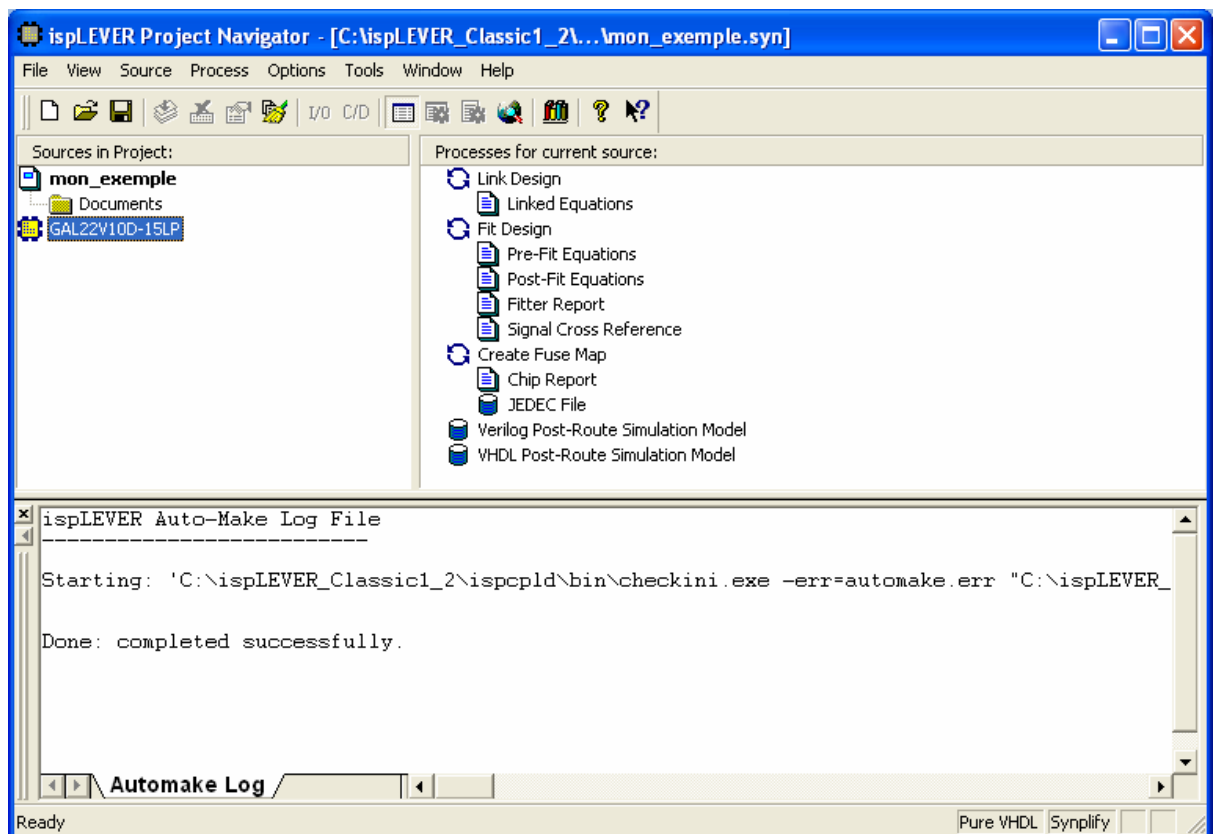


The 'Project Wizard - Select Device' dialog box is shown. The 'Family' list on the left is the same as the first dialog. The 'Device' list on the right includes 'GAL22V10D', 'GAL22V10D', 'GAL26CV12B', 'GAL26CV12C', 'GAL26V12C', 'ispGAL22LV10', 'ispGAL22V10AB', and 'ispGAL22V10AC'. The 'Speed grade: (ns)' is set to '-15', 'Package type' is '24PDIP', 'Operating conditions' is 'Commercial', and 'Power' is 'Low'. The 'Part Name' field shows 'GAL22V10D-15LP'. The 'Device Information' section on the right shows: Status: Production, Density: -, Logic cells: -, I/O cells: 10, I/O pins: 10, Dedicated input: 11, Output enable: 0, and Icc: 130 mA. At the bottom, there are checkboxes for 'Use I/O Assistant Flow' and 'Show Obsolete Devices', and buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

➔ Next

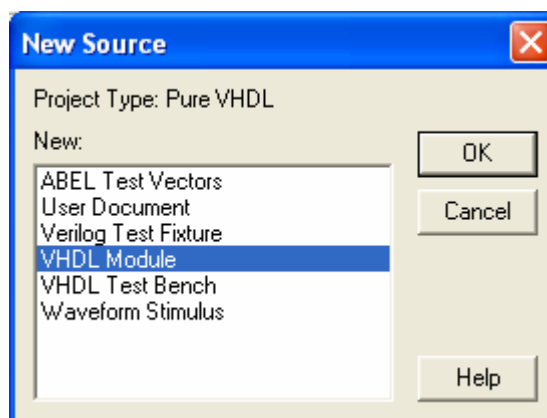
3. Insertion d'un fichier VHDL *description* existant

Copiez au préalable votre (vos) fichier(s) source dans le répertoire du projet. Ajoutez-les. → *Next* → *Finish*, vous retournez à la figure principale suivante. En cliquant sur *GAL22V10D-15LP*, vous trouverez la liste des processus réalisables. Un click droit suivi de *start* sur l'un d'eux, l'exécute.



4. Ajout d'un nouveau fichier VHDL *description*

Click sur *Documents* → *New...* → choisir *VHDL Module*



→ *OK* → introduire le nom du fichier (Counter.vhd), de l'entité (my_counter), de l'architecture (arch_my_counter) ainsi que les entrées/sorties du système (*CLK* en *in*, *RST* en *in*, *CNT_VAL* en *out* avec MSB à 3 et LSB à 0, i.e. il s'agit d'un vecteur de sortie de 4 bits).

New VHDL Source

File Name:

Entity:

Architecture:

Port Name	Direction	MSB	LSB
CLK	in		
RST	in		
CNT_VAL	out	3	0

OK Cancel Browse...

→ OK → l'éditeur de texte s'ouvre.

5. L'éditeur VHDL

Text Editor - [counter.vhd]

File Edit View Templates Tools Options Window Help

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

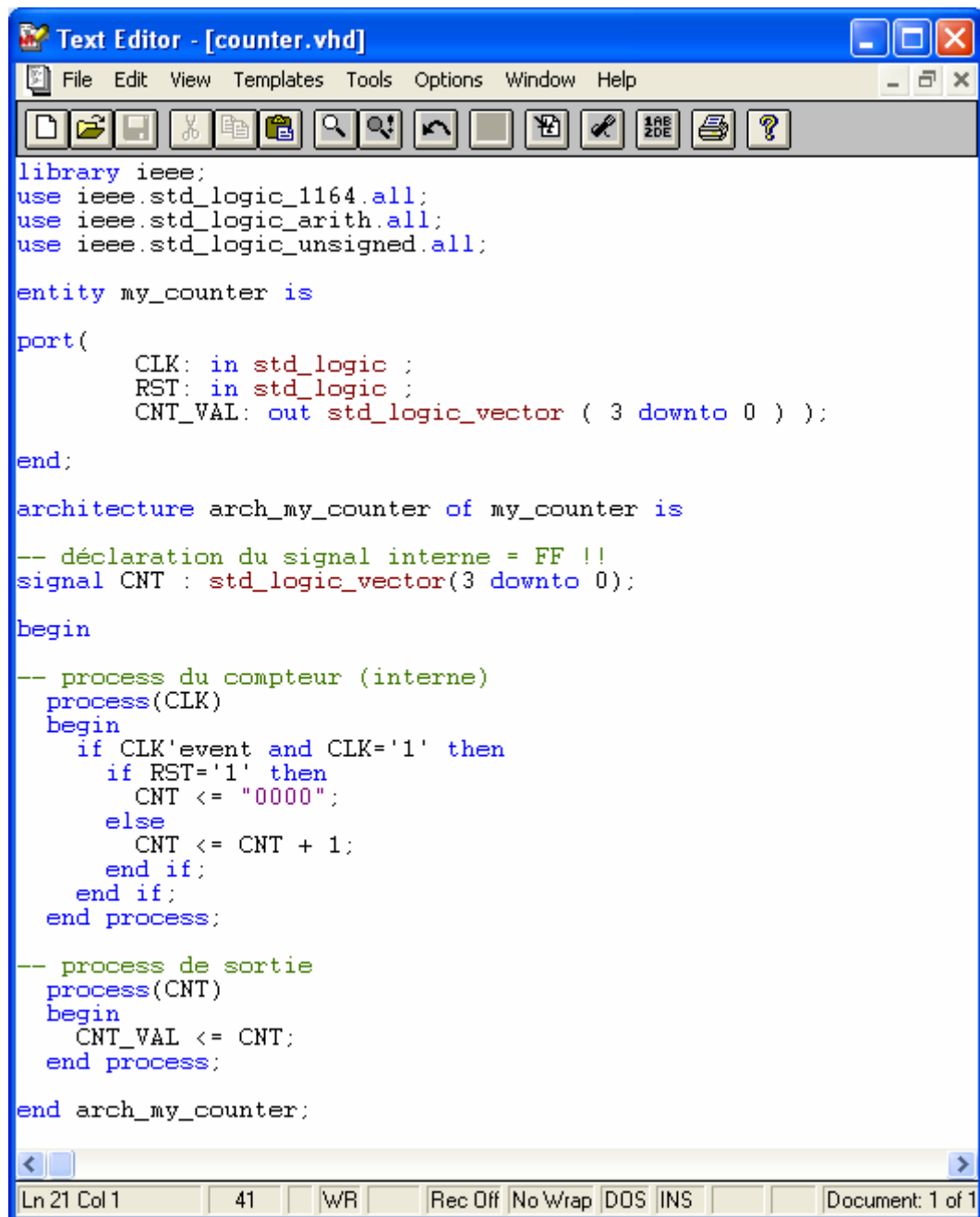
entity my_counter is
port(
    CLK: in std_logic ;
    RST: in std_logic ;
    CNT_VAL: out std_logic_vector ( 3 downto 0 ) );
end;

architecture arch_my_counter of my_counter is
begin
end arch_my_counter;

```

Ln 1 Col 1 19 WR Rec Off No Wrap DOS INS Document: 1 of 1

Il reste à compléter l'architecture



```
Text Editor - [counter.vhd]
File Edit View Templates Tools Options Window Help
[Icons]
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity my_counter is
port(
    CLK: in std_logic ;
    RST: in std_logic ;
    CNT_VAL: out std_logic_vector ( 3 downto 0 ) );
end;

architecture arch_my_counter of my_counter is
-- déclaration du signal interne = FF !!
signal CNT : std_logic_vector(3 downto 0);

begin
-- process du compteur (interne)
process(CLK)
begin
    if CLK'event and CLK='1' then
        if RST='1' then
            CNT <= "0000";
        else
            CNT <= CNT + 1;
        end if;
    end if;
end process;

-- process de sortie
process(CNT)
begin
    CNT_VAL <= CNT;
end process;

end arch_my_counter;
```

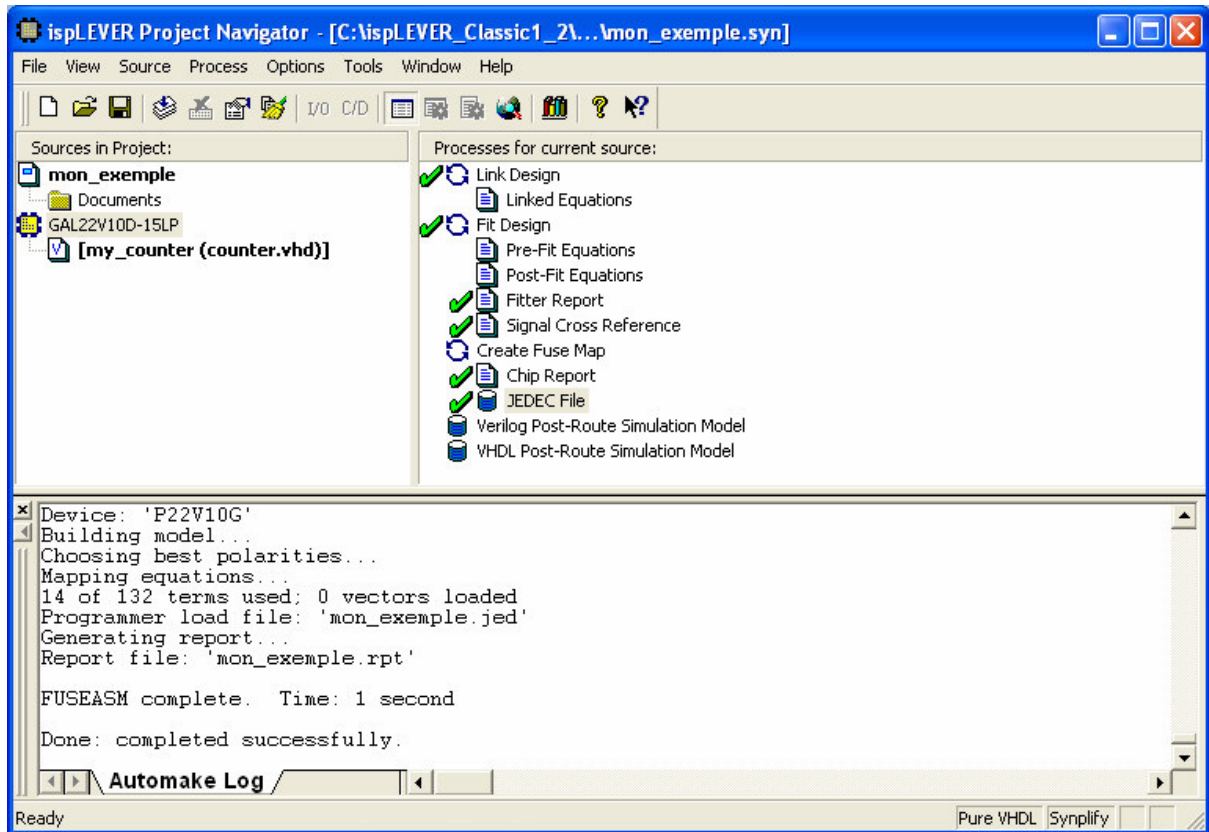
Ln 21 Col 1 41 WR Rec Off No Wrap DOS INS Document: 1 of 1

Le premier process met à jour les flip-flop interne, CNT. CNT n'est affecté que dans un et un seul process ! Chaque signal ou sortie ne peut être affecté que dans un seul process. Le second process se charge de connecter les FF aux pins de sortie.

Ici, il n'y a pas d'attribution des pins. Voir les notes de labo pour un exemple.

6. Compilation

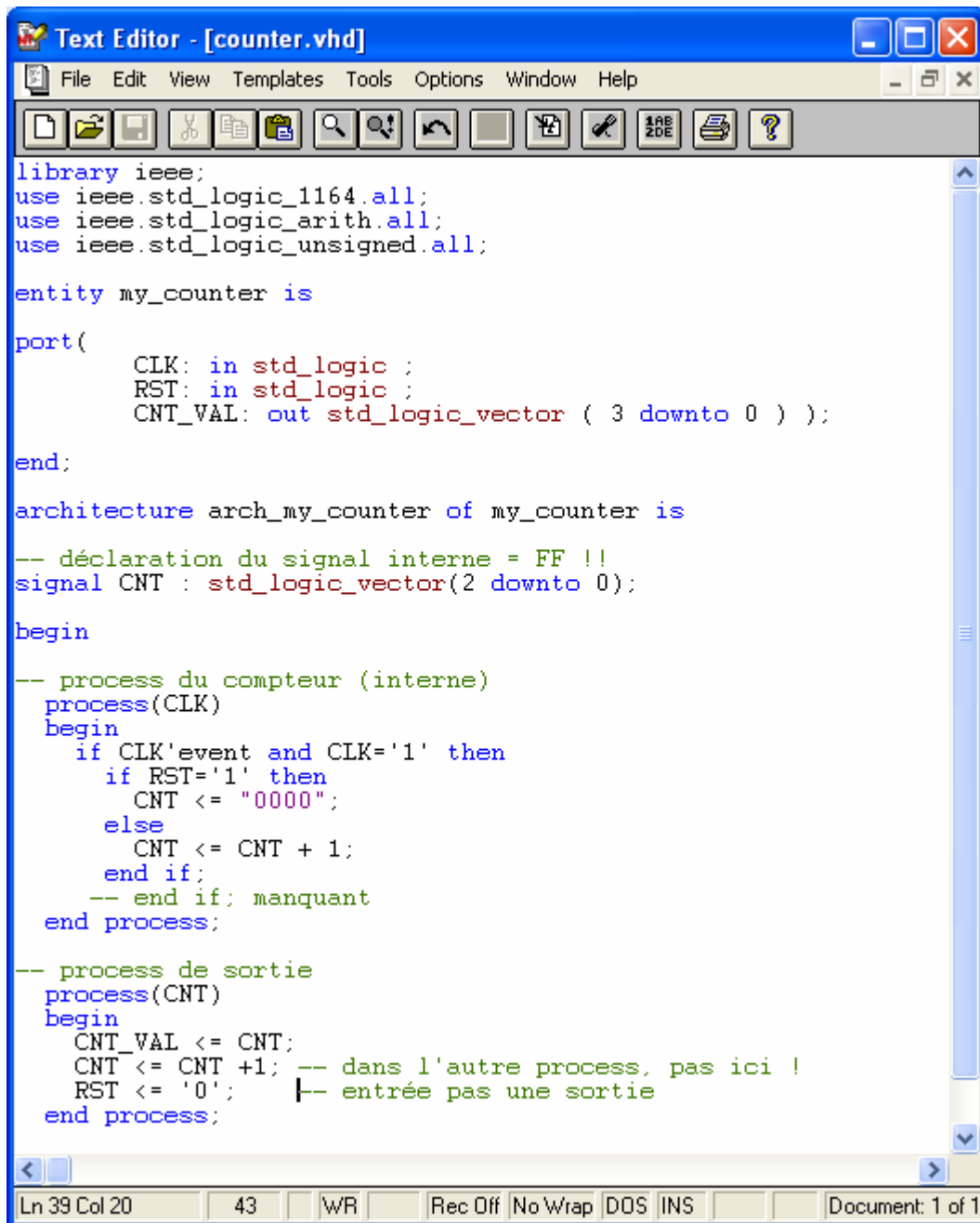
Allez dans la fenêtre principale, click sur le nom du composant (*GAL22V10D-15LP*) → click droit sur *JEDEC file* → *Start*.



Si tout se déroule sans erreur, la dernière phrase du rapport est *Done : completed successfully*.

7. Les erreurs

En cas d'erreurs, faites défiler le rapport pour atteindre le descriptif des erreurs.
Partons du code suivant



```
Text Editor - [counter.vhd]
File Edit View Templates Tools Options Window Help
[Icons]

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

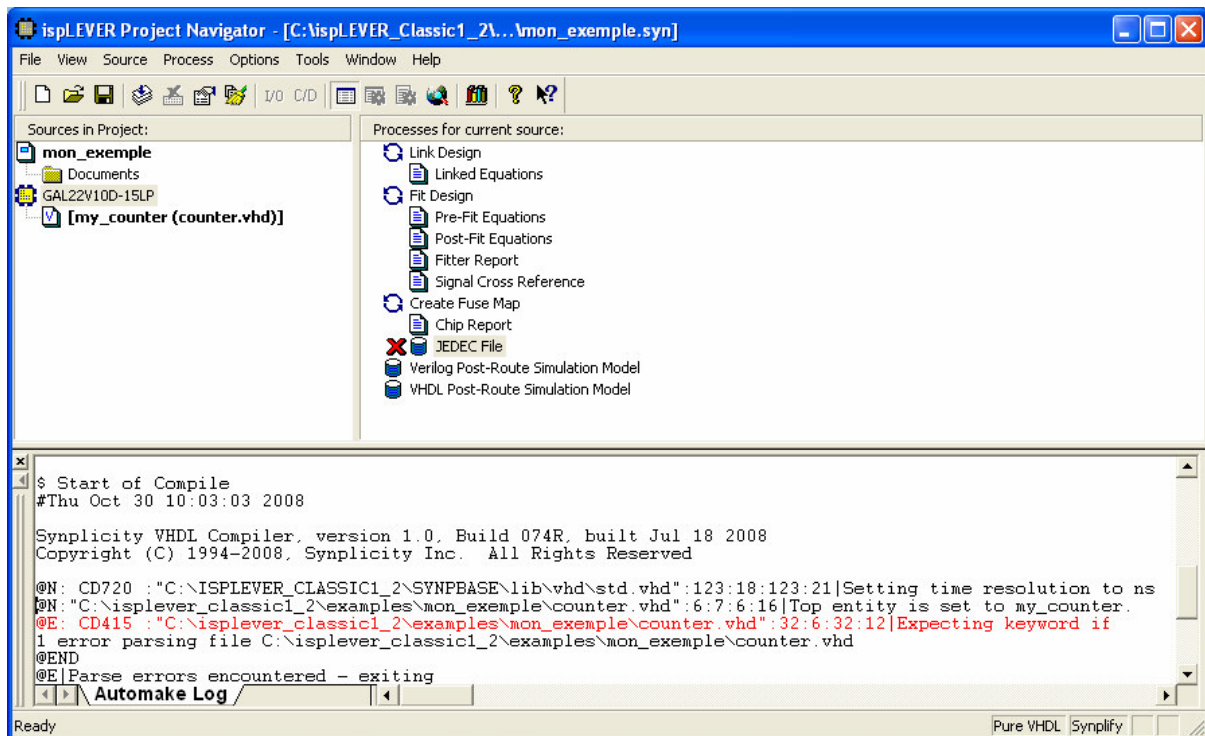
entity my_counter is
port(
    CLK: in std_logic ;
    RST: in std_logic ;
    CNT_VAL: out std_logic_vector ( 3 downto 0 ) );
end;

architecture arch_my_counter of my_counter is
-- déclaration du signal interne = FF !!
signal CNT : std_logic_vector(2 downto 0);
begin
-- process du compteur (interne)
process(CLK)
begin
    if CLK'event and CLK='1' then
        if RST='1' then
            CNT <= "0000";
        else
            CNT <= CNT + 1;
        end if;
        -- end if; manquant
    end process;

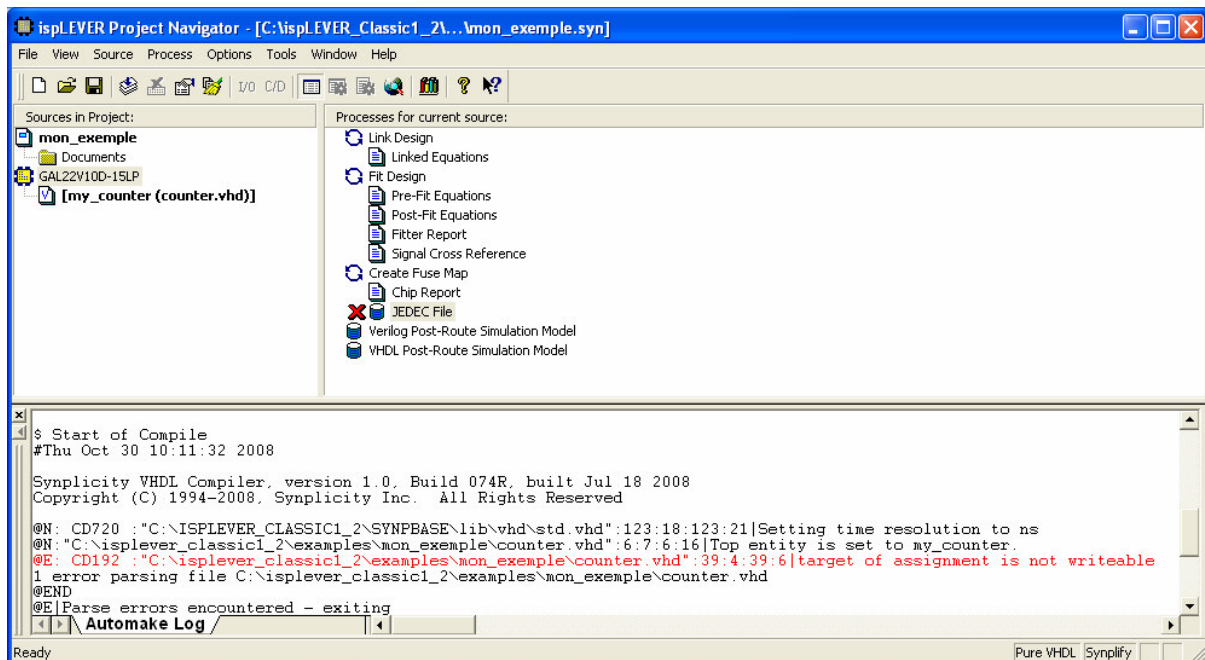
-- process de sortie
process(CNT)
begin
    CNT_VAL <= CNT;
    CNT <= CNT + 1; -- dans l'autre process, pas ici !
    RST <= '0';    -- entrée pas une sortie
end process;
```

Ln 39 Col 20 43 WR Rec Off No Wrap DOS INS Document: 1 of 1

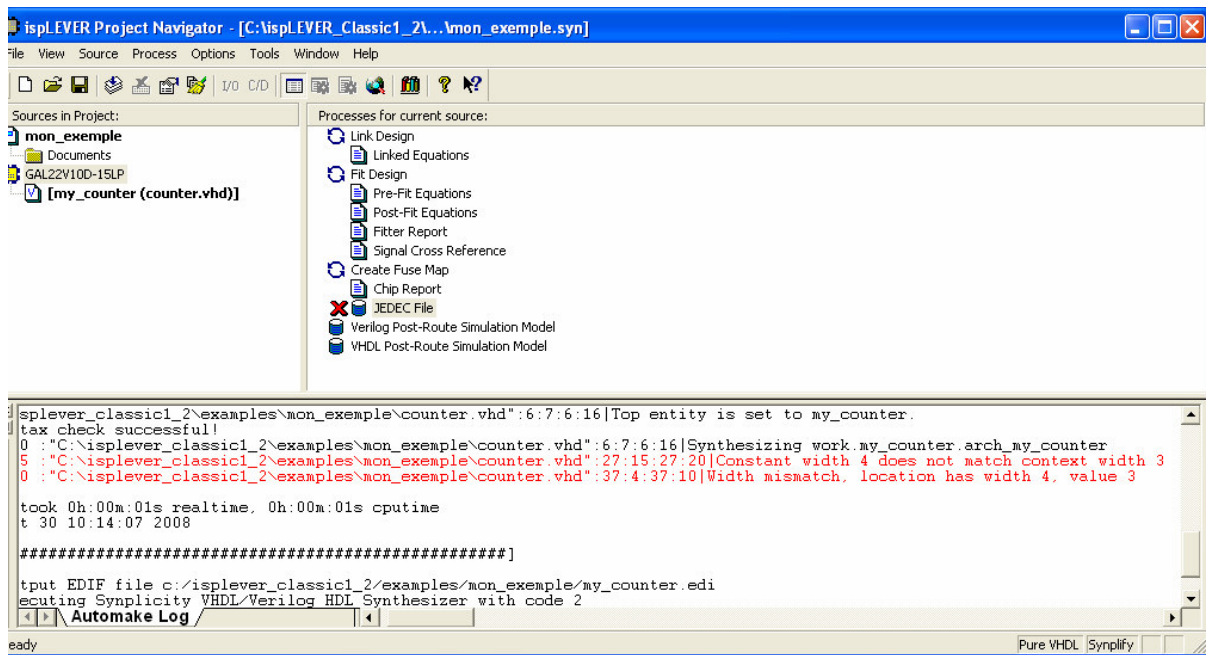
Lancez la compilation,



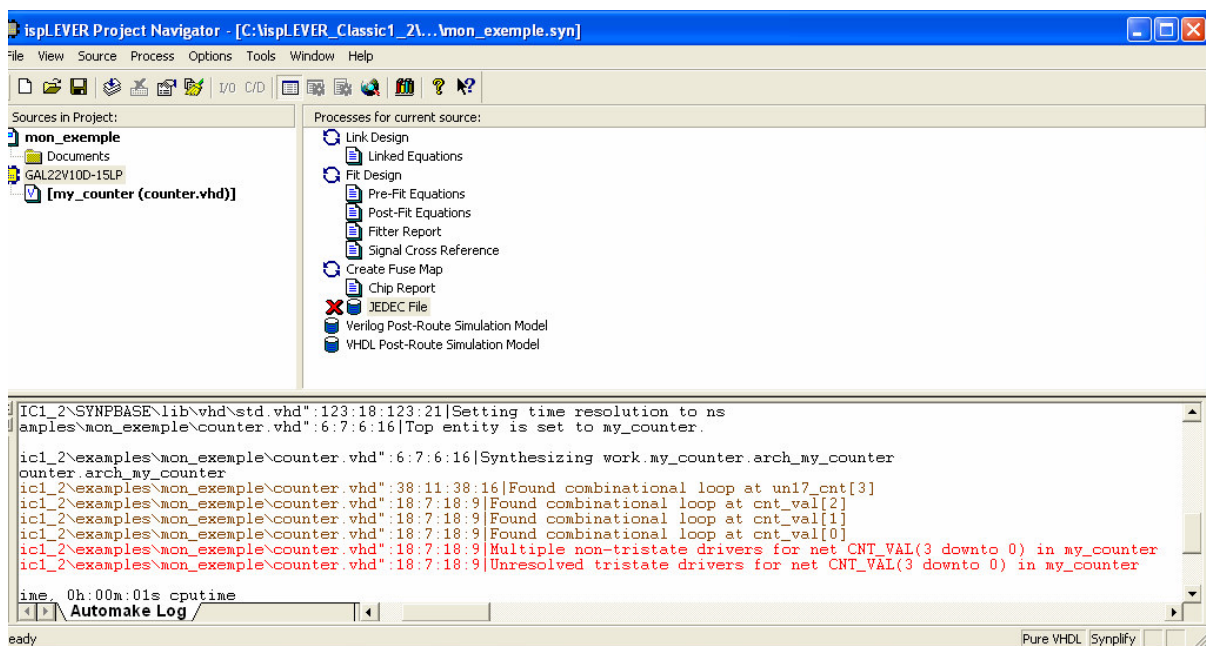
Il manque un if... Relancez la compilation



Ligne 39, RST ne peut pas être affecté. Corrigez, et relancez la compilation



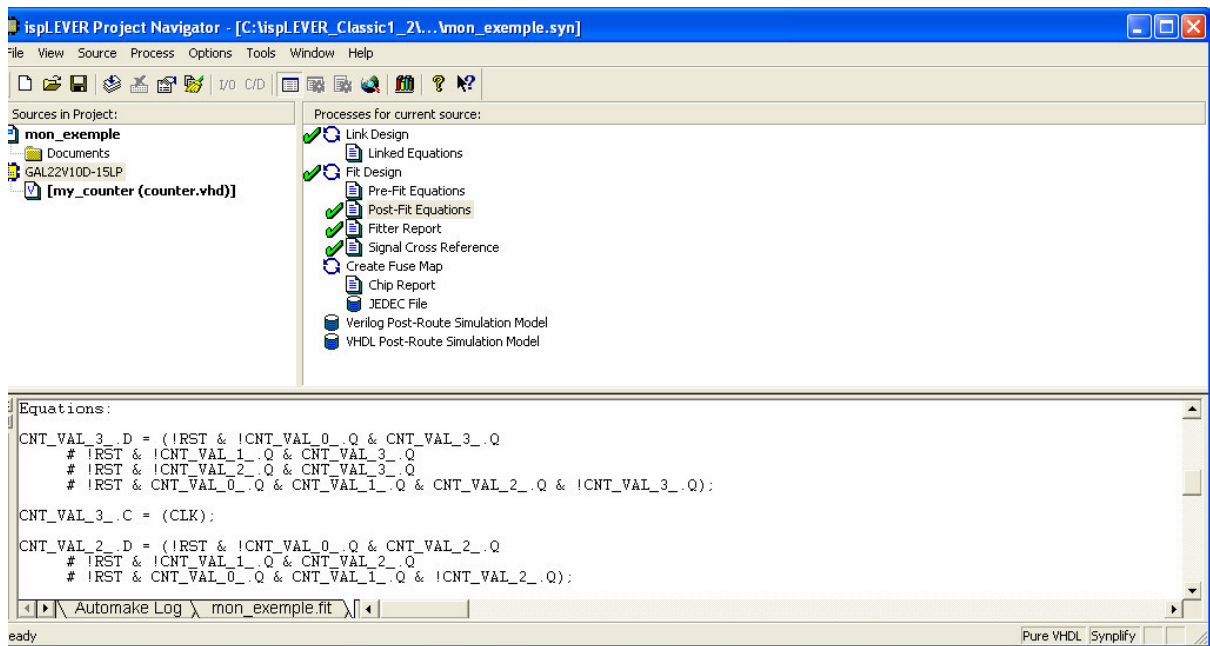
Le signal CNT est défini sur 3 bits alors que CNT_VAL est sur 4 bits. Remettez CNT sur 4 bits, relancez la compilation



CNT est affecté dans 2 process, ce qui est interdit matériellement. Un FF peut changer sur base d'une horloge et non de lui-même (surtout si c'est un vecteur de bits, car alors l'horloge est une fonction complexe). Cette erreur est l'exemple type d'une programmation orienté « software » et non « hardware ».

8. Les rapports

Vous pouvez visualiser les équations de sorties avec un click droit sur *Linked Equations* ou *Post-Fit Equations* → *Start*



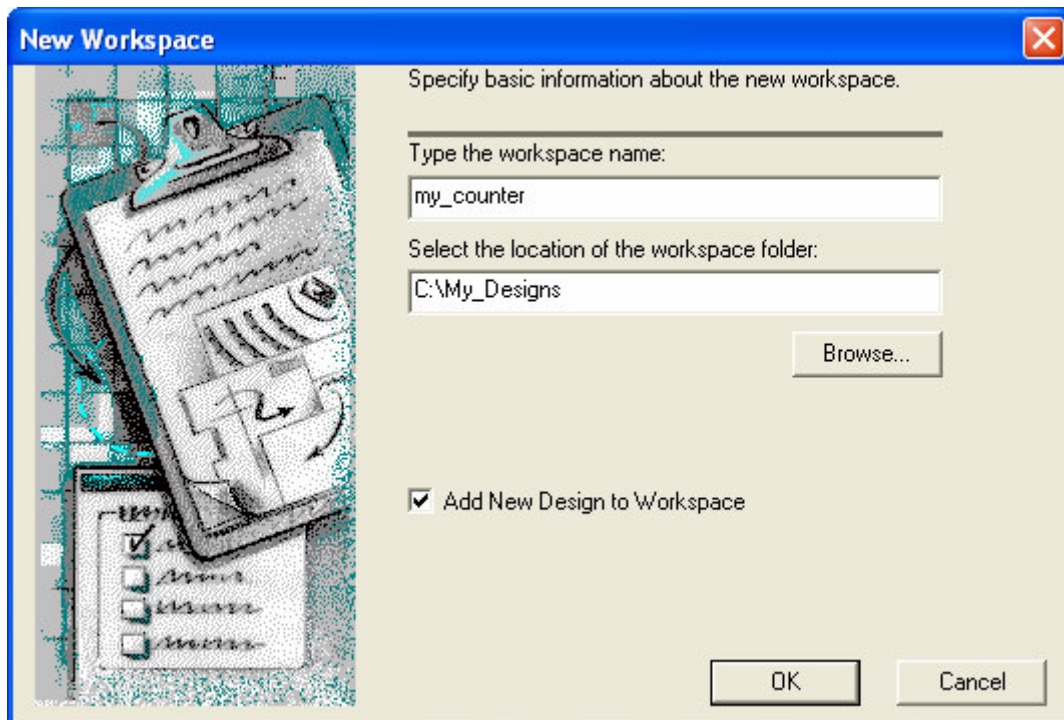
Quant au *Fitter report*, il vous donnera l'assignation des signaux d'entrée/sortie aux pins du composant.

Active-HDL 8.1LWE

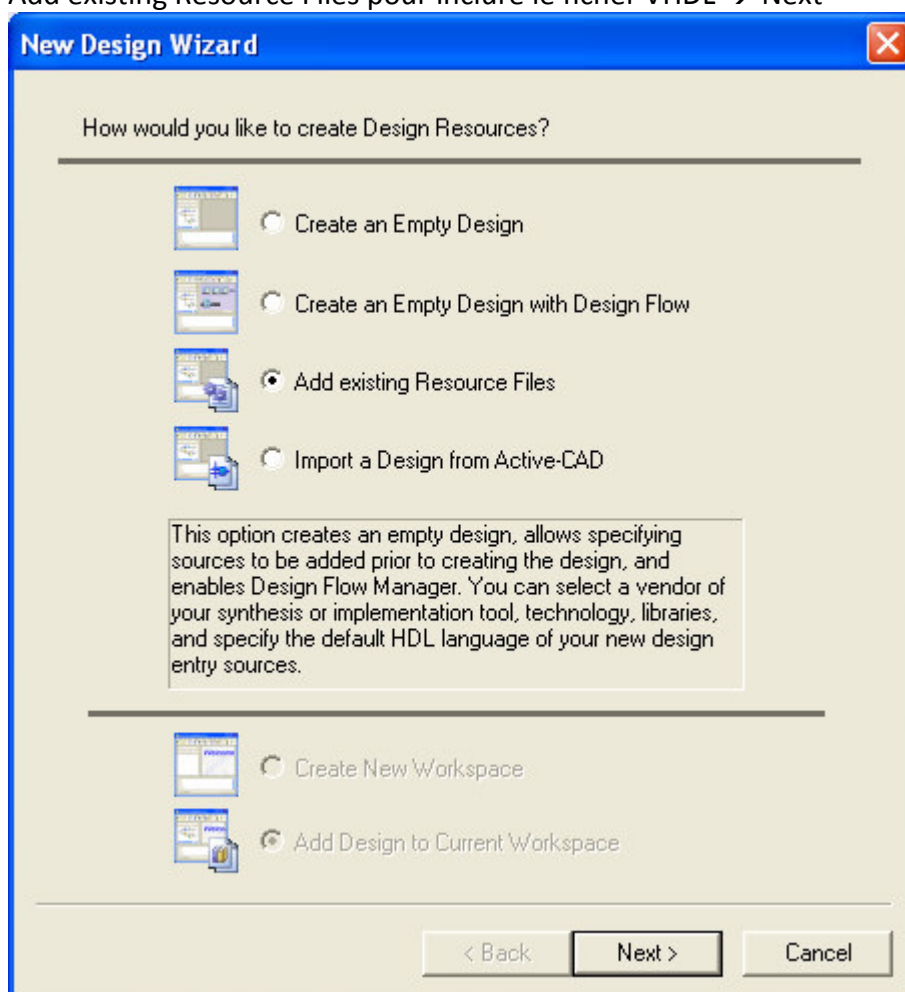
Cet outil de simulation vous aidera à valider le comportement de votre système décrit en VHDL. Vous devrez définir les stimuli d'entrée (horloges et entrées binaires) et vous aurez l'occasion de suivre pas à pas l'évolution temporelle des signaux internes et sorties du système. Il doit être clair que les résultats fournis par HDL-Sim ne sont qu'une version informatisée du comportement (exempt des délais inhérents aux portes logiques, flip-flops, défauts dans la carte, ...). Nous pourrions différencier le *comportement virtuel fourni par HDL-Sim* du comportement réel du composant programmable une fois placé dans le circuit électronique.

1. Création du projet de simulation

Nous allons simuler le compteur. *File* → *New* → *Workspace* → nommez-le my_counter par exemple → *OK*

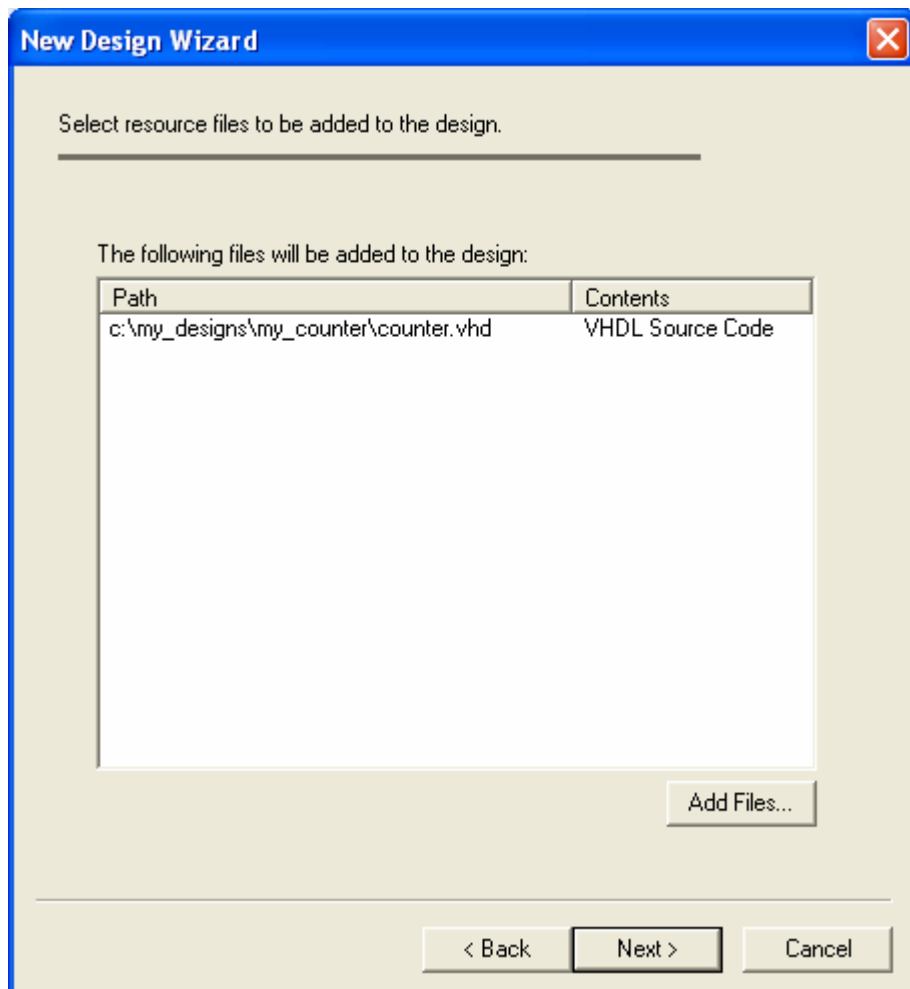


Add existing Resource Files pour inclure le fichier VHDL → Next



2. Inclusion des fichiers VHDL

Copiez le fichier VHDL dans le répertoire nouvellement créé (C:\My_Designs\my_counter\) et sélectionnez-le → *Next* → Choisissez le langage VHDL à la place de VERILOG → *Next* → donnez un nom (my_counter) au design → *Next* → *Finish*



New Design Wizard

Specify basic information about the new design.

Type the design name:

my_counter

Select the location of the design folder:

c:\My_Designs\my_counter

Browse...

The name of the default working library of the design:

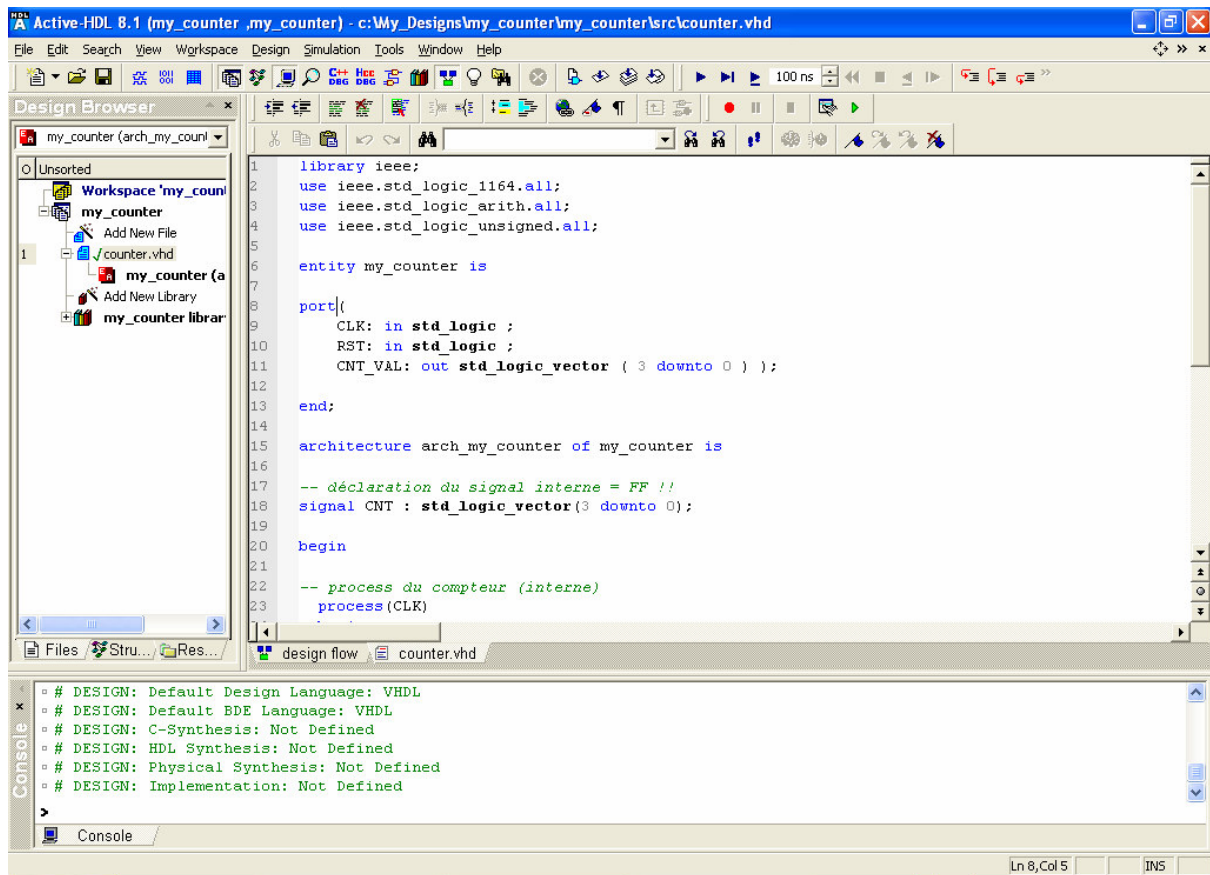
my_counter

The name specified here will be used as the file name for the library files and as the logical name of the library. You can change the logical name later on.

< Back

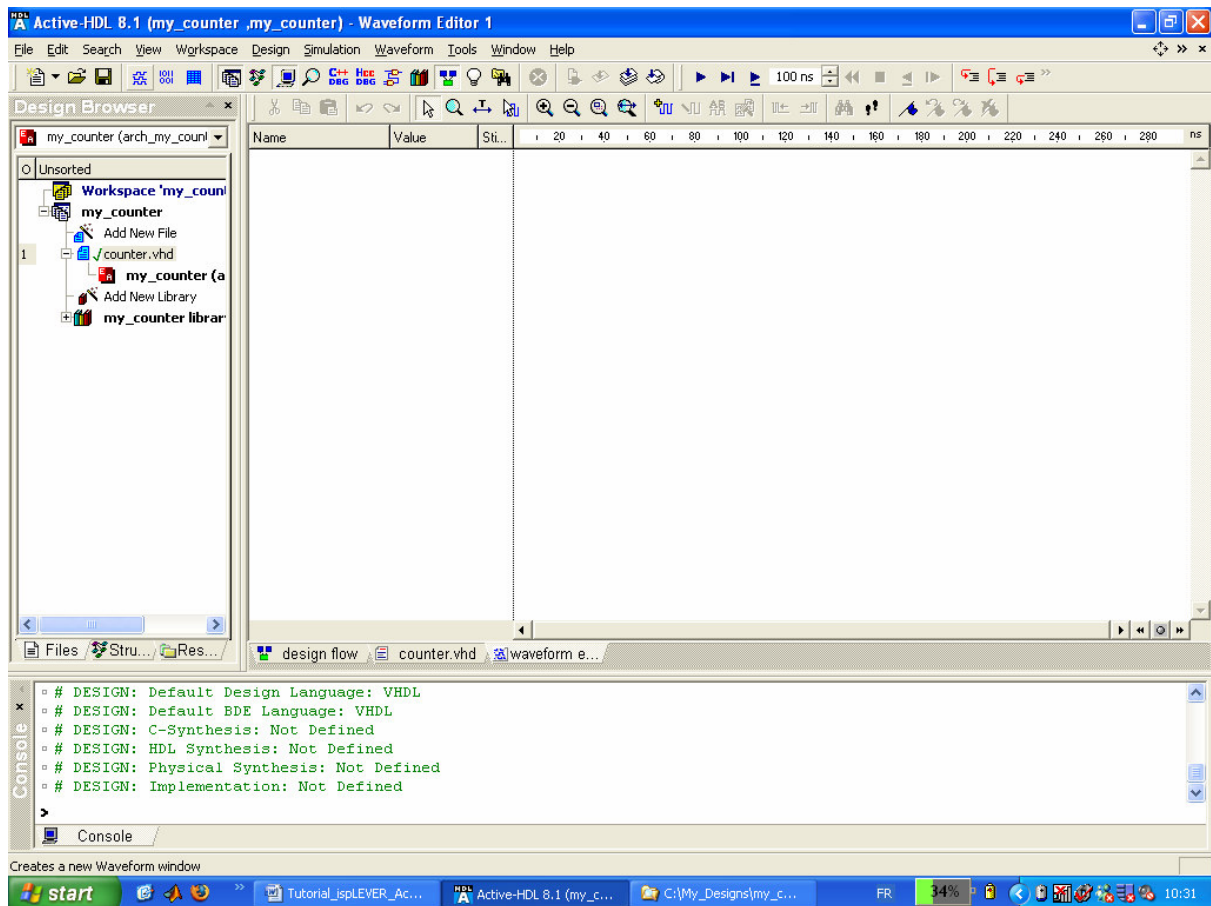
Next >


Cancel

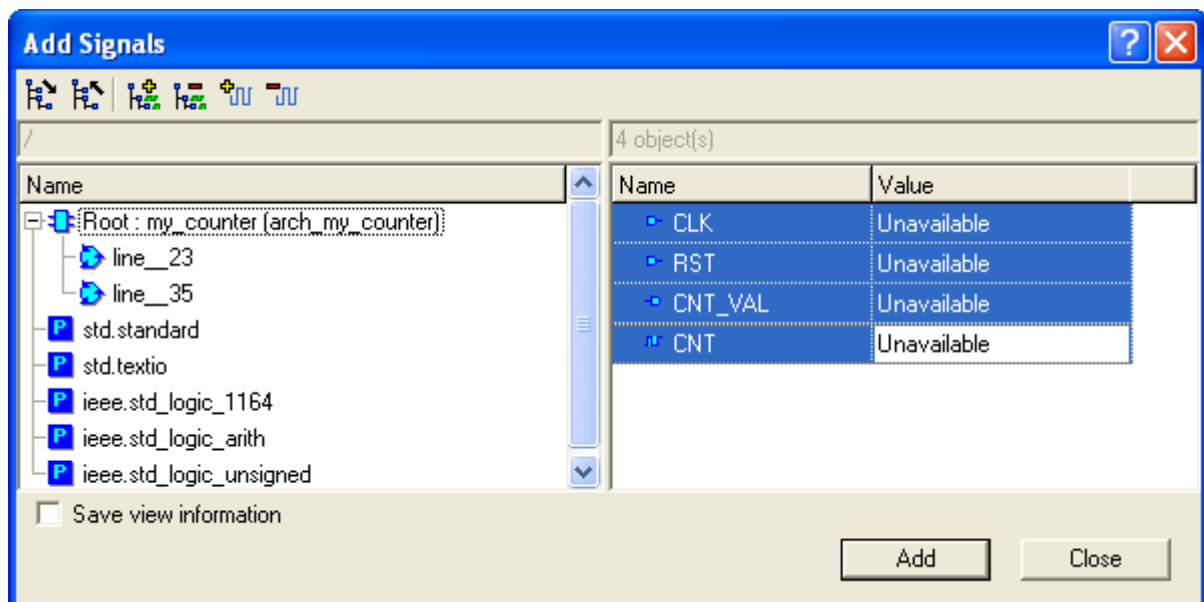



3. Edition des stimuli

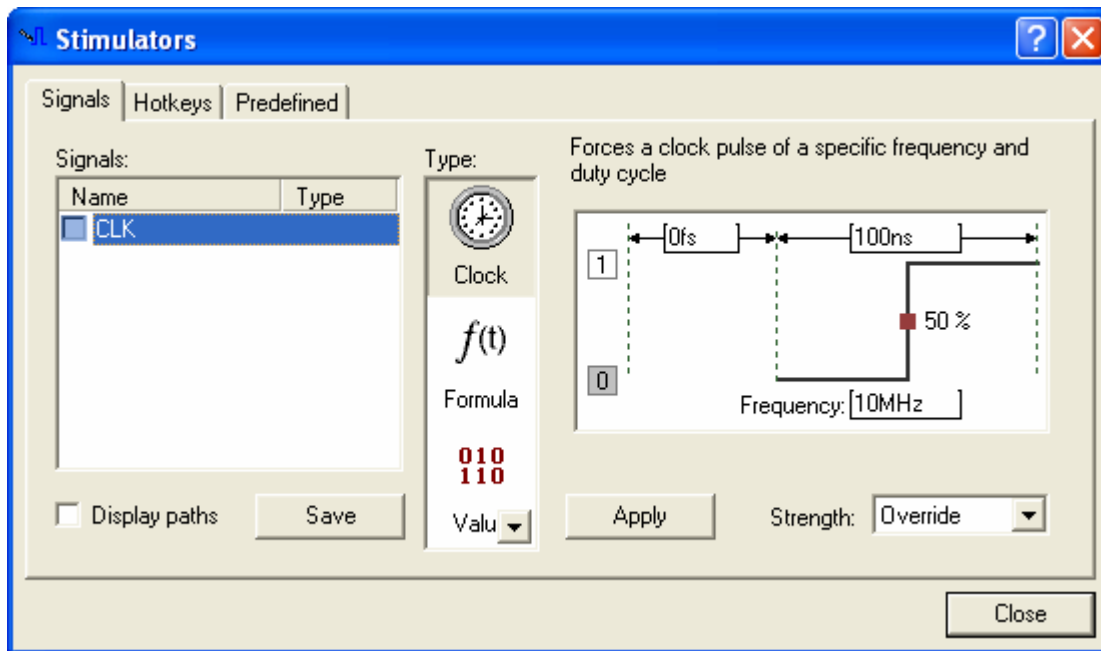
Cliquez sur  pour ouvrir une nouvelle simulation



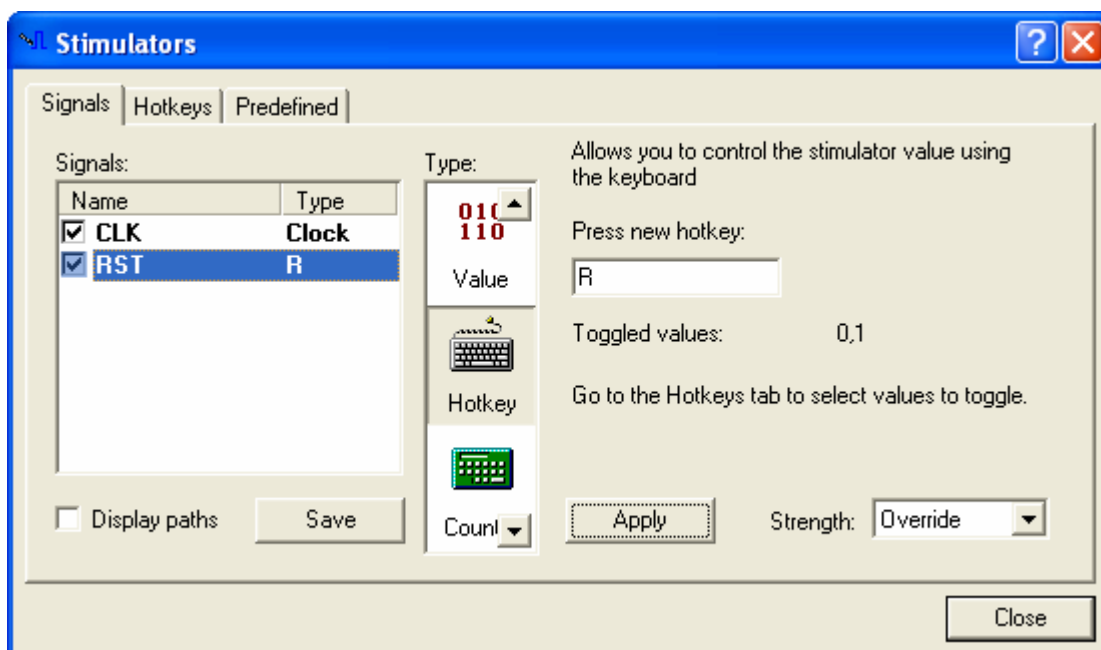
Cliquez sur  pour ajouter les signaux d'entrée/sortie



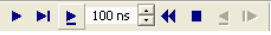

Ensuite, sélectionnez une entrée (CLK) et cliquez sur . Assignez-lui un type Horloge et cliquez sur *Apply*.

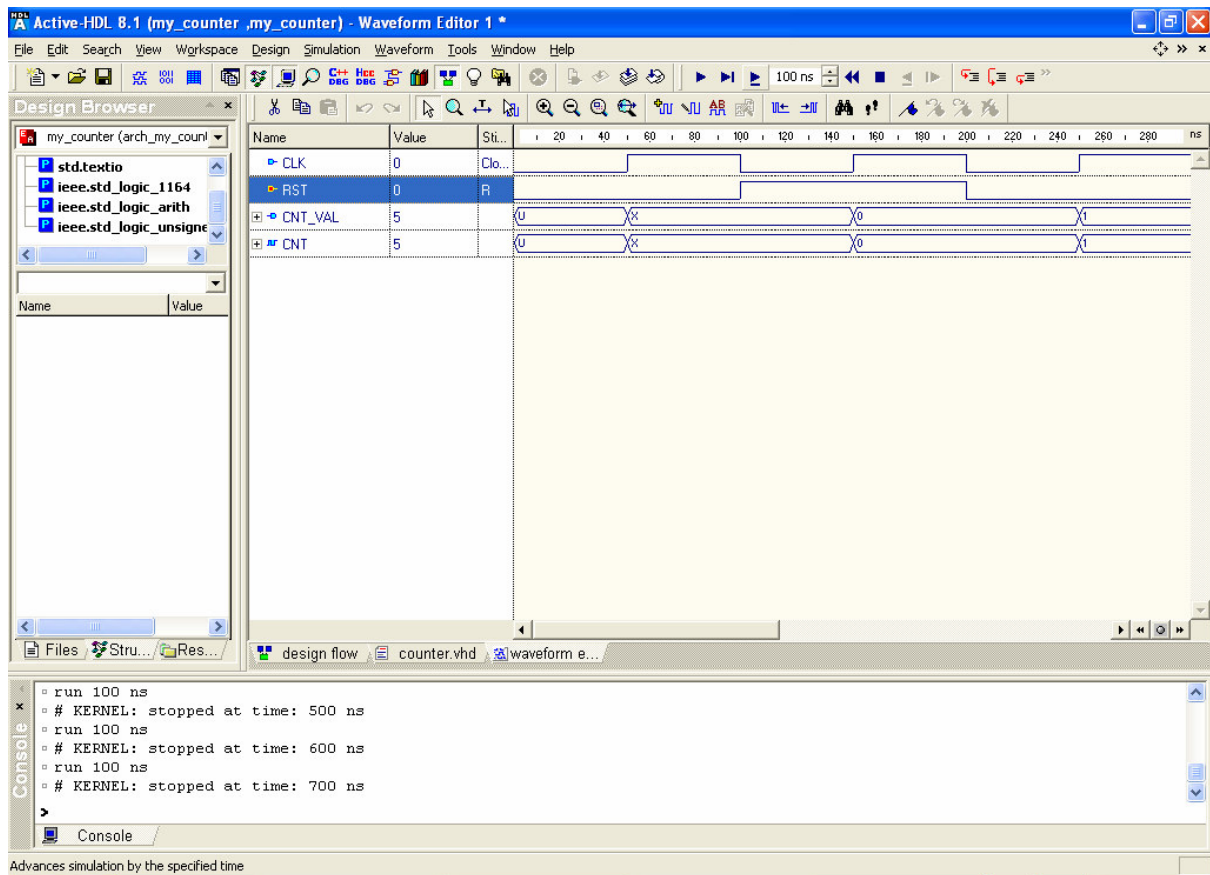


Faites de même avec RST mais assignez-lui un hotkey, la touche R (cliquez dans le champ « Press new hotkey » puis tapez la touche R), cliquez sur *Apply*.



4. Simulation

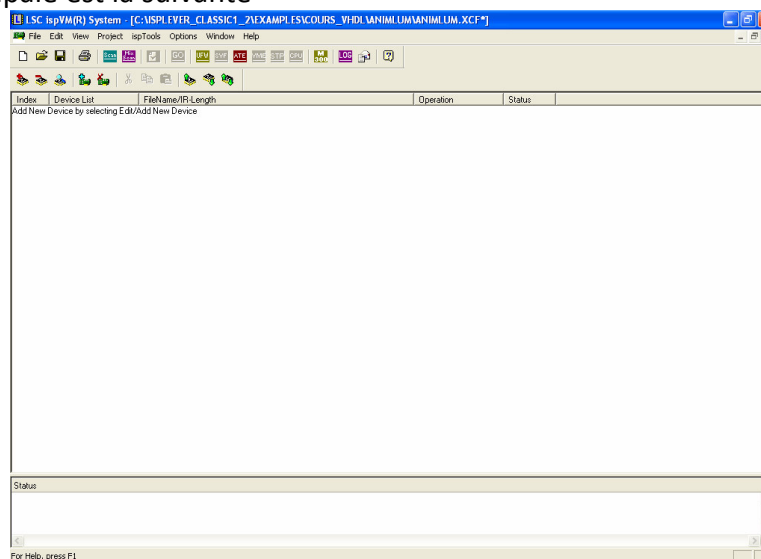
Cliquez sur  pour gérer la simulation. Le premier symbole lance la simulation sur un temps très long. Le plus simple est d'utiliser le 3^{ème} () pour simuler pas à pas, e qui vous permet d'appuyer sur la touche R et de changer la valeur de l'entrée RST. Après quelques click sur le bouton vous obtenez la simulation suivante




ispVM System

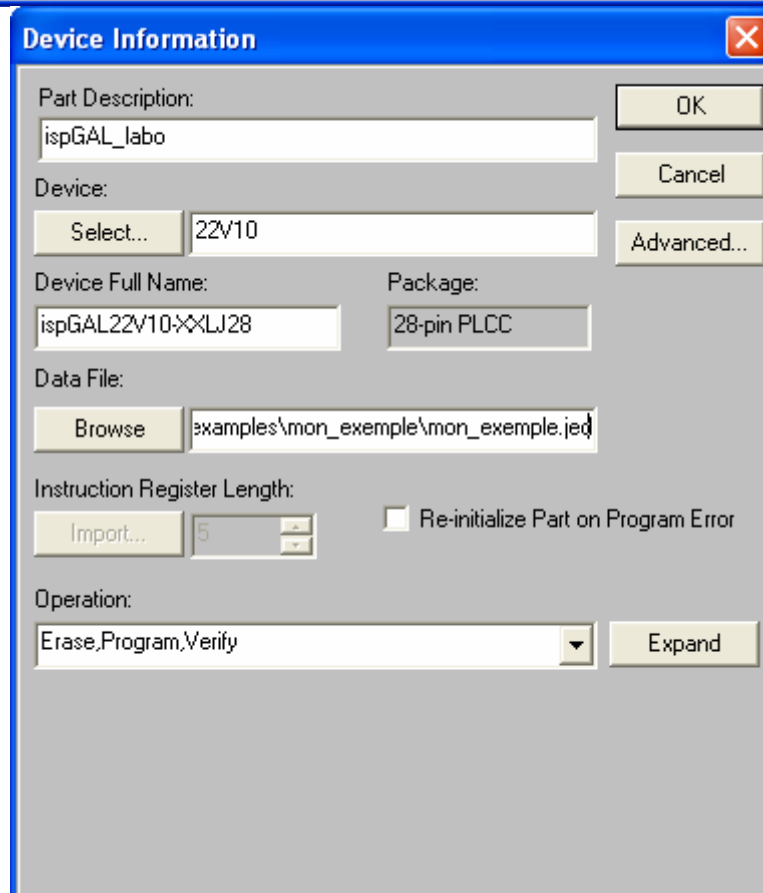
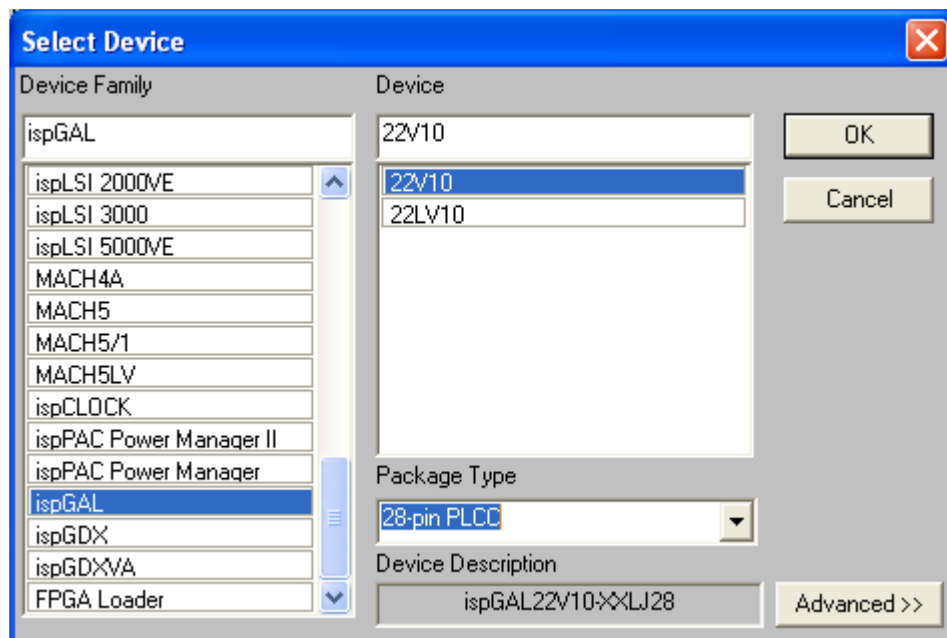
L'outil ispVM System vous permet de programmer un composant in situ, i.e. directement sur la carte comme c'est le cas de l'ispGAL22V10C de la carte de laboratoire. Une autre solution est le dataman48PRO+ qui peut programmer un très grand nombre de composants dont le package est DIL ou DIP (le package du compteur asynchrone par exemple est DIP alors que la GAL est en PLCC).

La fenêtre principale est la suivante



1. Ajout du fichier JEDEC

Cliquez sur  pour ajouter un composant à programmer (ayant l'extension *isp* comme l'*ispGAL22V10*) → *Select...*, complétez le champ Part Description et sélectionnez le fichier JEDEC (*mon_exemple.jed*).



2. Le câble USB n'est pas reconnu !

Options → Cable I/O and port setup et assurez-vous que le type de câble est bien USB.

Cable and I/O Port Setup

Cable Type: Auto Detect

☒ Port Setting:

☐ Custom Port:

OK Cancel Debug Mode

NOTE: Auto Detect only works with the Lattice parallel port and USB cables.

NOTE: Connecting the board's TRST pin to the cable's TRST pin is not recommended. Instead, connect the board's TRST to Vcc. The parallel port on some PCs may not be able to hold a high during the programming, and may have glitches, interrupting programming.

NOTE: When using ORCA devices (JTAG Mode) with the Lattice parallel port cable, connect the ispEN pin to SERIAL connector PROG pin.

☐ TRST/Reset Pin Connected
☐ Set High
☐ Set Low


☒ ispEN/BSCAN Pin Connected
☐ Set High
☒ Set Low

☐ DONE Pin Connected

☐ PROGRAMN Pin Connected

☐ INITN Pin Connected

3. Programmation du composant

Vous obtenez la fenêtre suivante. Cliquez sur 

LSC ispVM(R) System - [C:\USPLEVER_CLASSIC1_2\EXAMPLES\COURS_VHDL\ANIMLUM\ANIMLUM.XCF*]

File Edit View Project ispTools Options Window Help

Download(Ctrl+G)

Index	Device List	FileName/IR-Length	Operation	Status
1	22V10	C:\usplever_classic1_2\examples\nron_exemple\nron_exemple.jed	Erase Program/Verify	N/A

Status

Download(Ctrl+G)