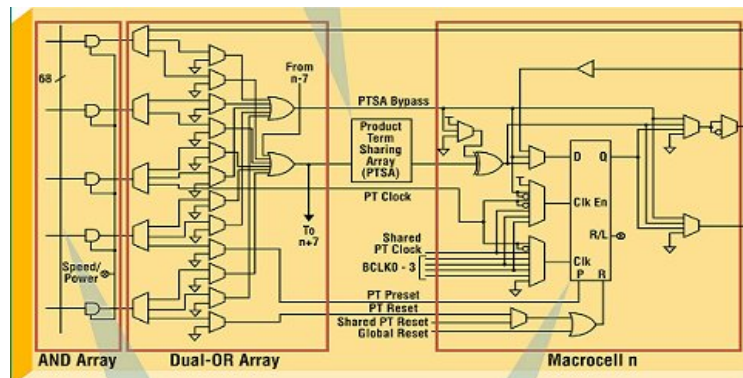


# ispLEVER pour les nuls



# Sommaire

<b>1</b>	<b>LOGICIEL.....</b>	<b>3</b>
1.1	TELECHARGER ISPLEVER .....	3
1.2	OBTENIR UNE LICENCE. ....	3
<b>2</b>	<b>PROJET .....</b>	<b>4</b>
2.1	DEMARRER ISPLEVER.....	4
2.2	CREER UN PROJET .....	4
2.3	CHOISIR LE COMPOSANT A PROGRAMMER : .....	5
<b>3</b>	<b>SOURCES.....</b>	<b>7</b>
3.1	AJOUTER UNE SOURCE AU PROJET : .....	7
3.1.1	<i>Ajout d'un composant.....</i>	8
3.1.2	<i>Ajouter des fils de connexions .....</i>	8
3.1.3	<i>Donner un nom à un potentiel :.....</i>	8
3.1.4	<i>Placer des entrées et sorties :.....</i>	9
3.1.5	<i>Création d'un symbole :.....</i>	9
3.1.6	<i>Erreur de connexion.....</i>	9
3.1.7	<i>Attribution d'une pin du composant :.....</i>	11
3.2	ASSOCIE UN FICHIER HDL A UN BLOC .....	12
3.3	HIERARCHIE DU PROJET .....	12
3.4	SIMULATION FONCTIONNELLE .....	13
3.4.1	<i>Ajout d'une source Abel Test vectors .....</i>	13
3.5	SIMULATION TEMPORELLE .....	14
3.6	AJOUT D'UNE SOURCE DE STIMULIS GRAPHIQUES POUR UNE SIMULATION.....	16
<b>4</b>	<b>PROGRAMMATION D'UN COMPOSANT .....</b>	<b>18</b>
4.1	GENERATION DU FICHIER JDEC .....	18
4.2	CHOIX D'UN FICHIER JDEC ET D'UN CIRCUIT A PROGRAMMER .....	18

Ce document a été construit pour une première utilisation d'ISP lever.

La version utilisée est la 2.0, à la date de l'édition du document il existe la version 4.0.

La carte électronique utilisée est starter\_kit (voir document) vendu par lattice sous la référence : [MACH4-SK44](#)

# ISPLever pour les nuls

## 1 LOGICIEL

Condition de téléchargement : liaison haut débit,

Condition d'utilisation : obtenir une licence temporaire de 6 mois (numéro de la carte réseau).

Pour obtenir le numéro de la carte réseau : après avoir installer ispLever , dans le menu lancer, ispLever registration. Des informations sur la carte réseau sont affichés.

### 1.1 Telecharger ispLever

Se connecter sur le site de lattice <http://www.latticesemi.com/>

Menu : product → Development tools → Software → ispLEVER starter → Downloadable Software

Telecharger ispLever starter Primary module (+ de 100Mo)

S'enregistrer "Create a new account" compléter les champs obligatoire et valider.

### 1.2 Obtenir une licence.

Menu : product → Development tools → Licensing

Rentrer votre email et password (voir enregistrement)

Rentrer le n°de la carte réseau **Enter Your Host NIC (Physical Address)**:

Donner l'email **Enter Your Email Address**:

Et valider [Generate license](#)

Télécharger depuis votre boîte aux lettres license.dat.txt

Placer la licence dans le répertoire isptools/license/license.dat

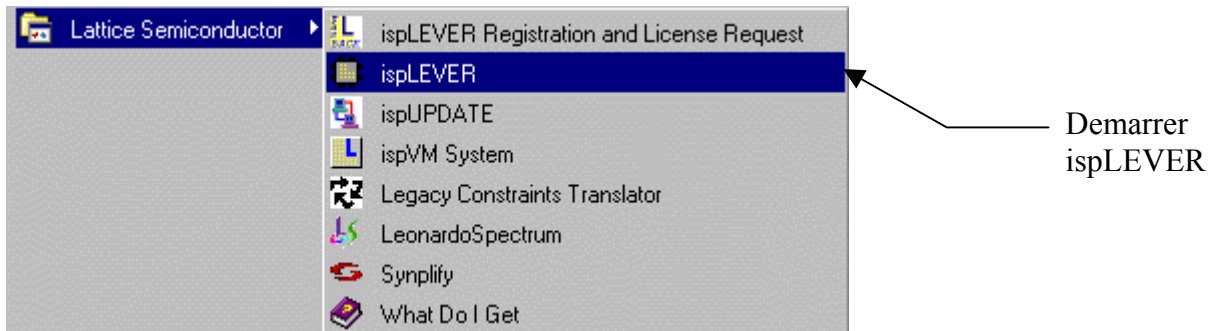
Le logiciel peut être utilisé.

La licence est valable 6 mois. Il faut donc recommencer au bout de 6 mois.

## 2 PROJET

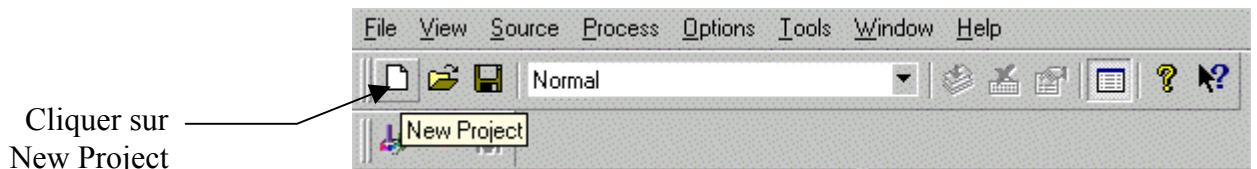
### 2.1 Démarrer ispLEVER

Lancer le logiciel *ispLEVER* : dans le menu *programme Lattice Semiconductor ispLEVER*



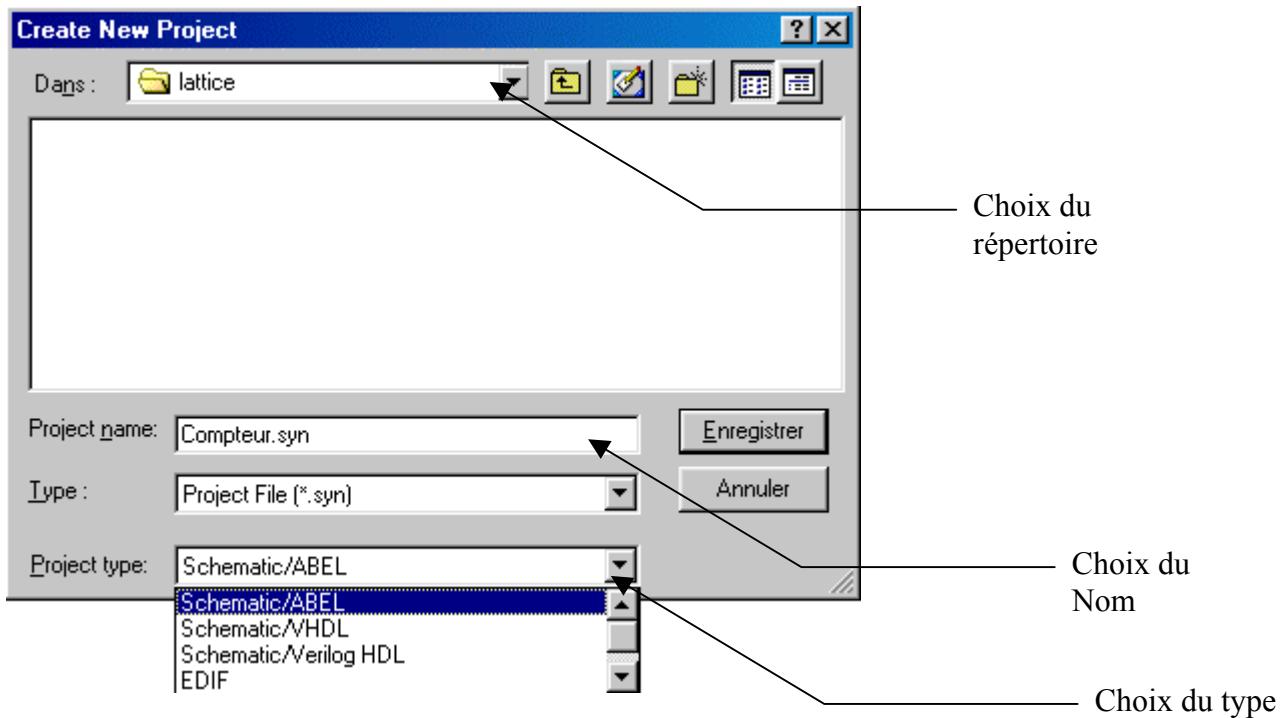
### 2.2 Créer un projet

Dans le menu *File New Project* ou *Ctrl+N* ou l'icône *New Project*



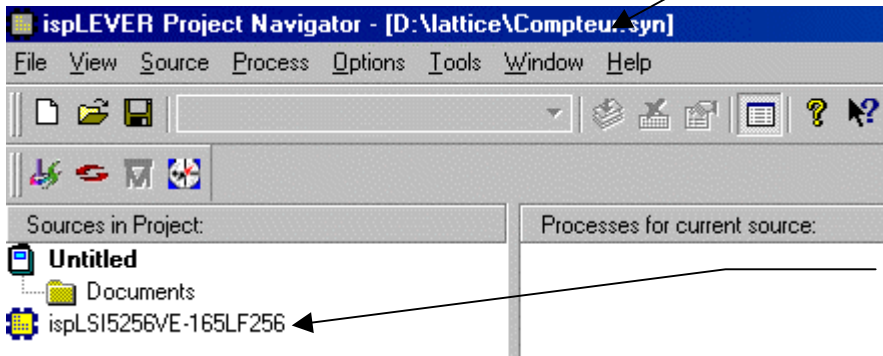
Enregistrer le projet :

- Choisir le répertoire, choisir le nom.
- Choisir le type de projet Schematic/ABEL, Schematic/VHDL, Schematic/Verilog HDL, EDIF, GDF.



Résultat des opérations :

Chemin et nom du projet



Circuit par défaut

### 2.3 Choisir le composant à programmer :

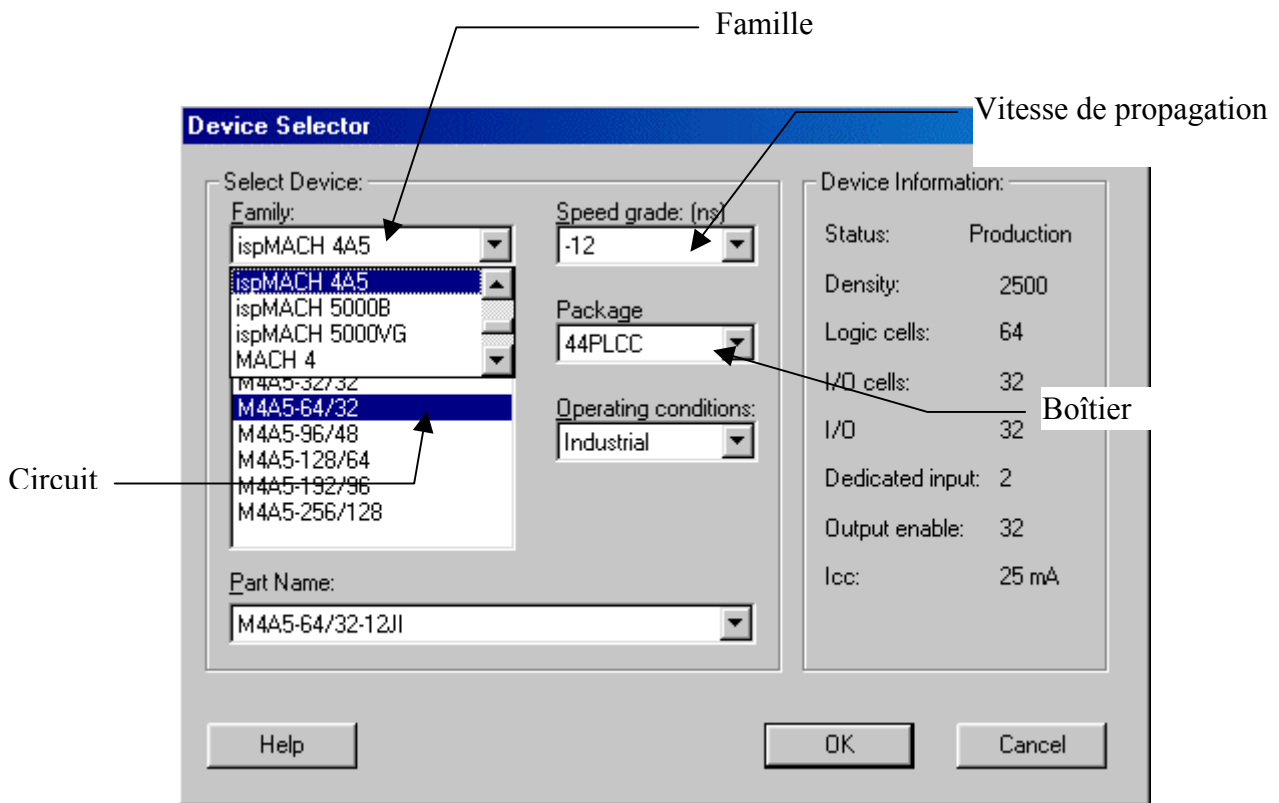
Click droit *Select New Device*



1- Click droit

2- Select new Device

Choisir le composant adéquat : famille, circuit, boîtier, vitesse de propagation.



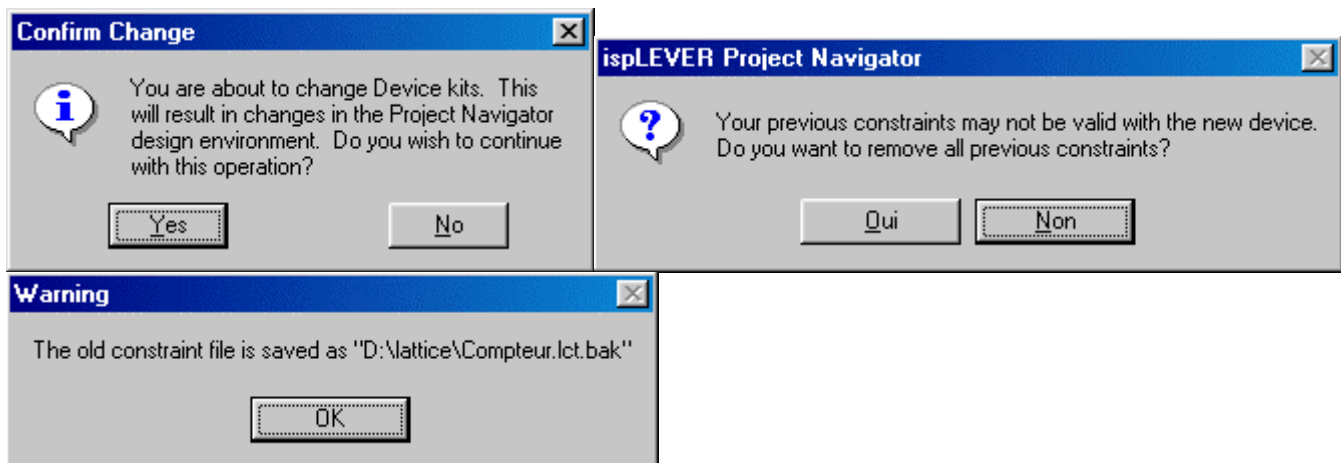
Famille

Vitesse de propagation

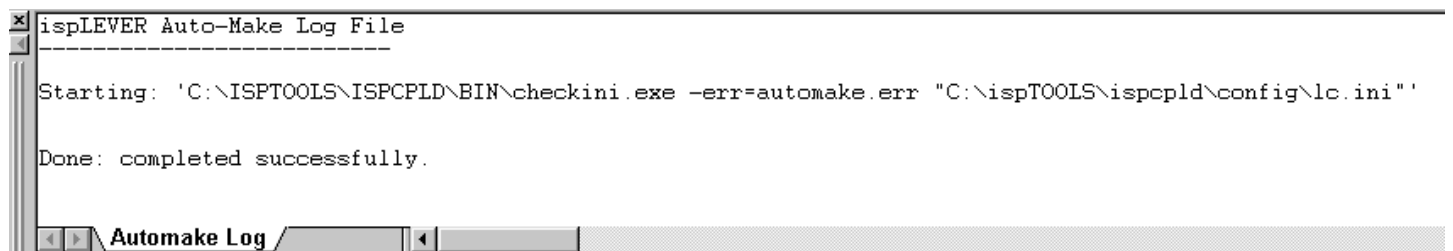
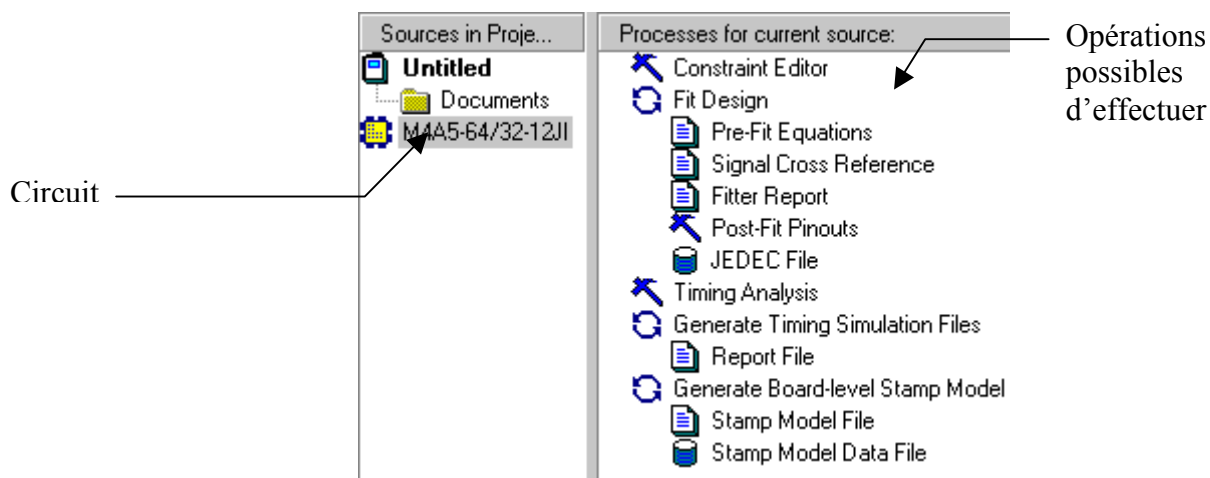
Boîtier

Circuit

Messages de confirmation :



Projet Final :



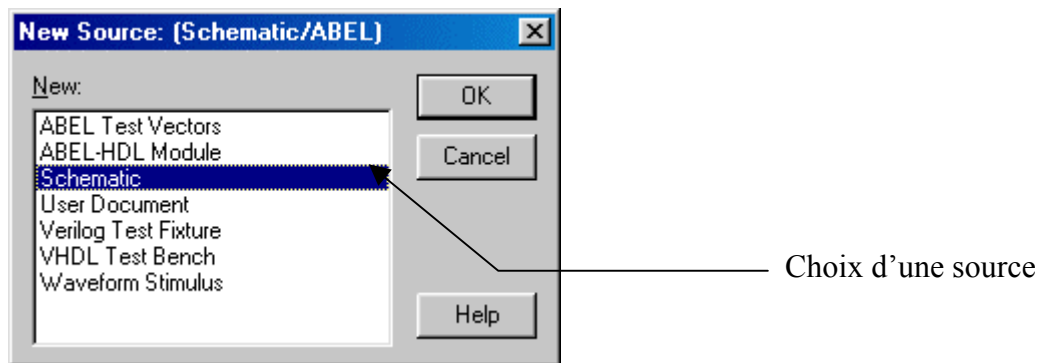
## 3 SOURCES

### 3.1 Ajouter une source au projet :

Clic droit sur le circuit et *new*

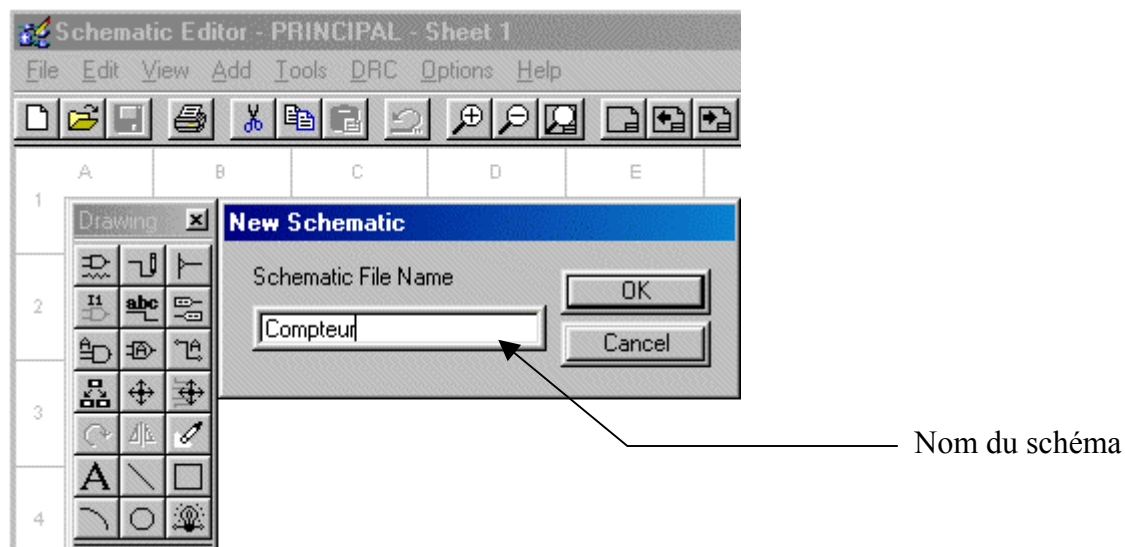


Ajout d'une source de type schéma, programme ou chronogramme (simulation uniquement).

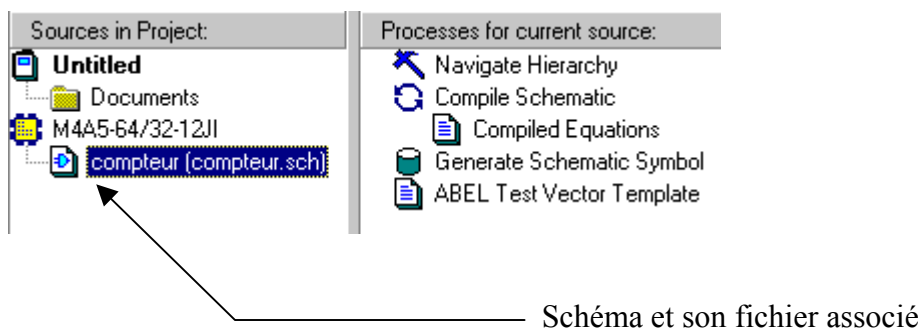


### Exemple pour une source schematic


Nommer le schéma.



Dans le projet, le schéma  apparaît avec l'extension .sch.



### 3.1.1 AJOUT D'UN COMPOSANT

Dans le menu Add, Symbol ou F2 ou click sur l'icône  de la barre des taches de dessins

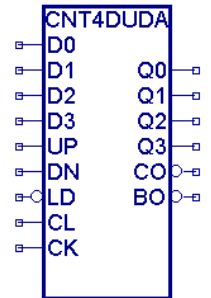


On ajoute un composant dans l'une des bibliothèques proposées.


Parmi les bibliothèques on trouve gate.lib (portes) regs.lib (bascules), iopads.lib (connectique vers les entrées sorties du composant), muxes.lib (multiplexeurs), arith.lib (additionneurs), title.lib et misc.lib (cartouches) vanttl.lib (quelques fonctions références ttl), (fonctions diverses : comparateurs, compteurs decodeurs, encodeurs, additionneurs, latches, inverseur 8 bits, multiplexeurs, détecteur de parité, registre à décalage), vanprim.lib (portes avec de nombreuses entrées) pll.lib (pll).

Pour l'exemple on choisit la bibliothèque vanfunc.lib et la fonction CNT4DUBA (Compteur 4 bits Decimal Up Down A).

C'est un compteur (UP)ou décompteur (DN) déclenchement sur front montant (CK) avec entrées (D0, D1, D2 & D3) de préchargement synchrone (LD) et asynchrone (CL). CO est la retenue du compteur.

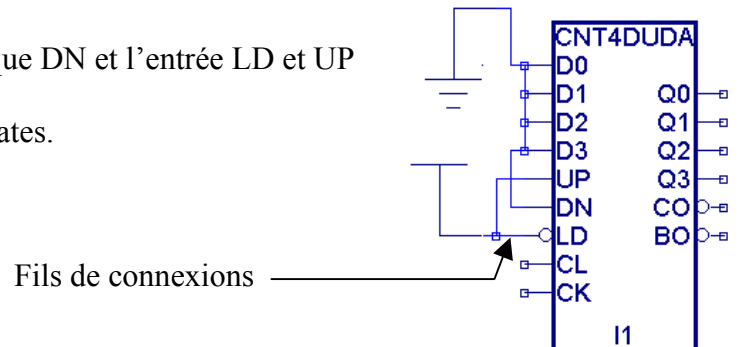


### 3.1.2 AJOUTER DES FILS DE CONNEXIONS

Dans le menu Add Wire ou F3 ou click sur l'icône Add Wire  de la barre des taches de dessins on ajoute des fils de connexions.


On reliera les entrées D0 à D3 à la masse ainsi que DN et l'entrée LD et UP à +VCC.

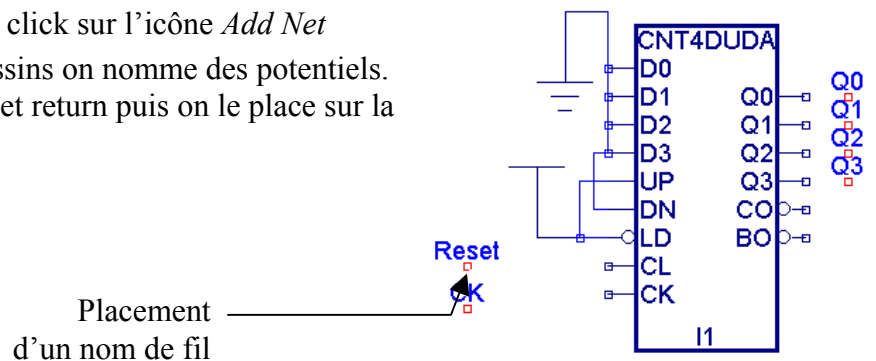
La masse et + VCC se trouve dans la librairie Gates.




### 3.1.3 DONNER UN NOM A UN POTENTIEL :

Dans le menu Add Net Name ou F4 ou click sur l'icône Add Net

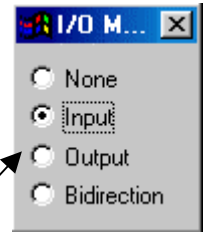
Name  de la barre des taches de dessins on nomme des potentiels. Dans la fenêtre du Bas on tape le nom et return puis on le place sur la feuille par un click.




### 3.1.4 PLACER DES ENTREES ET SORTIES :

Dans le menu *Add I/O Marker* ou *alt M* ou click sur l'icône *Add I/O Marker*  de la barre des taches de dessins on place des entrées sorties.

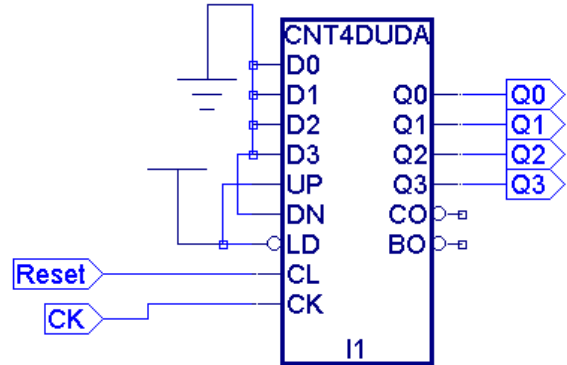
Une fenêtre permettant de choisir le type d'entrée sortie apparaît  
Choisir une sélection parmi les 4 proposés et cliquer sur le petit rectangle rouge associé au nom.  
Q0 à Q3 sont des sorties et CLK et RESET sont des entrées




Choix d'un type d'entrée sortie

Câbler les étiquettes  aux entrées sorties correspondantes.

Le schéma final est le suivant :

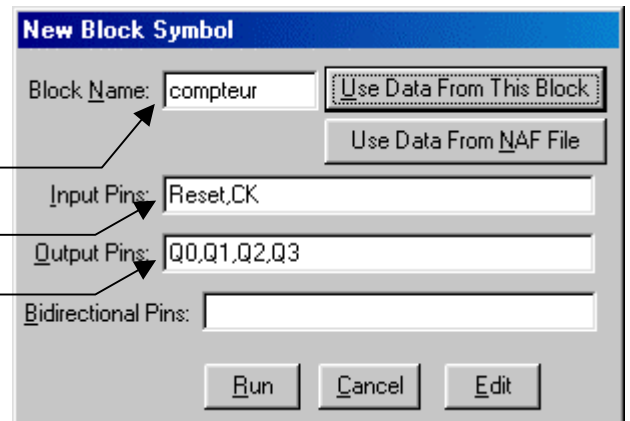



### 3.1.5 CREATION D'UN SYMBOLE :

Dans le menu *Add New Block Symbol* ou click sur l'icône *Add Block Symbol*  de la barre des taches de schematic editor, on associe le schéma à un symbole.


Dans la fenêtre on clique sur *Use Data From This Block* (cela permet d'associer directement les entrées sorties à ce bloc) puis sur *Run*.

Nom du bloc  
Listing des entrées  
Listing des sorties



On peut visualiser ce nouveau composant dans la librairie  Local symbol compteur.

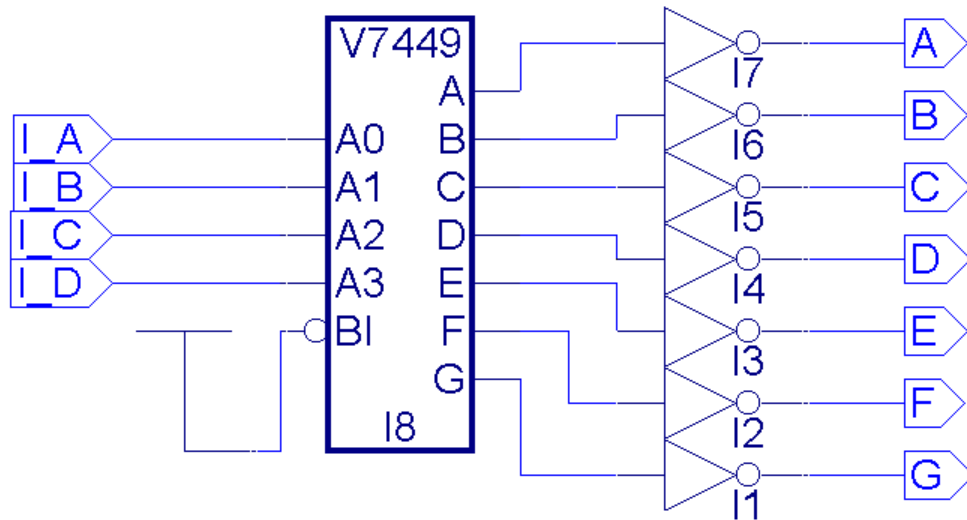
### 3.1.6 ERREUR DE CONNEXION

Dans le menu *DRC Consistency Check* ou click sur l'icône *Consistency*  de la barre des taches de schematic editor, on vérifie les éventuelles incohérence du schéma électrique.  
Une fenêtre error report communique les erreurs.

Fermer les fenêtres de l'éditeur de schéma.


**Créer un nouveau schéma (voir §3.1) appelez le transcodeur.** Ajouter le composant V7449 dans la librairie vantt.lib, puis le composant G\_INV dans la librairie gates.lib.


On peut se servir des outils dupliquer , déplacer  et effacer  de la barre des taches de dessins, afin de réaliser un transcodeur actif en sortie à l'état bas. Réaliser le schéma suivant :



Créer un symbole  pour ce schéma.

**Créer un nouveau schéma (voir §3.1) appelez le Principal.**

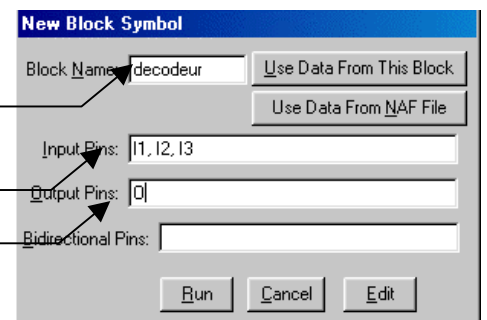
Ajouter les deux composants créés depuis la librairie Local  (symbol compteur & transcodeur).

Créer un symbole  sans correspondance de source

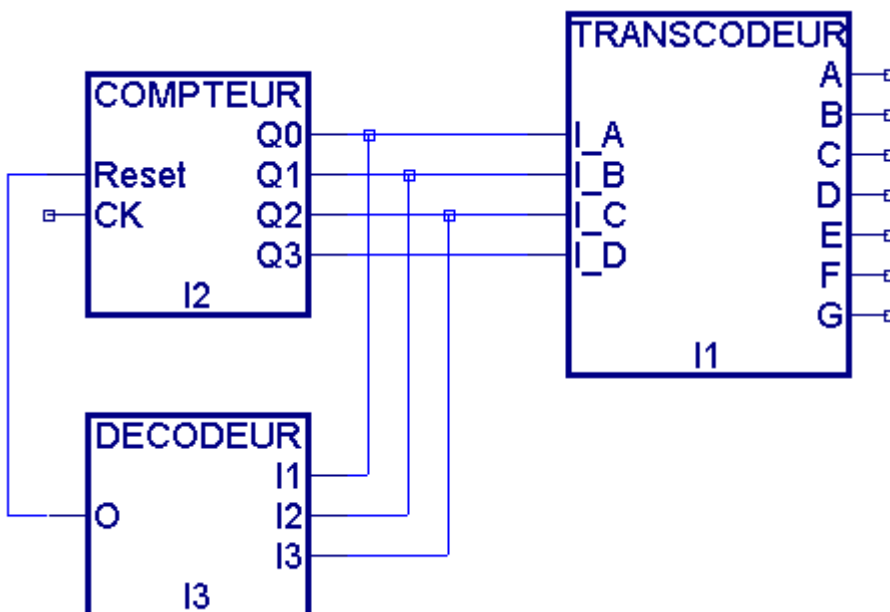
Nom : Decodeur


Entrées : I1, I2, I3

Sorties : O

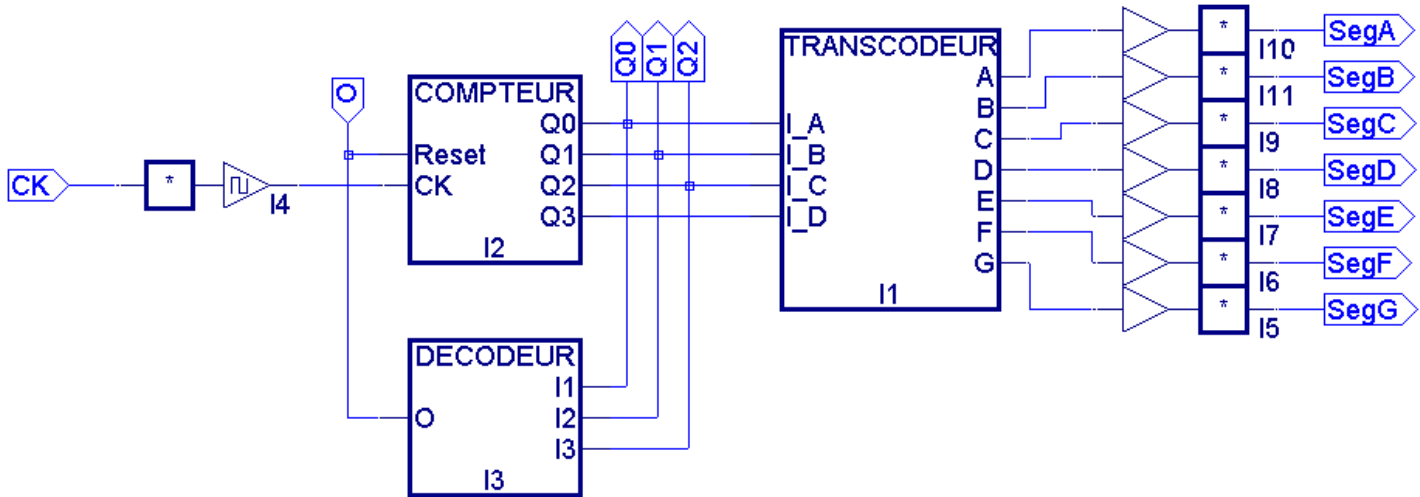


Placer le sur le schéma principal et effectuer les liaisons suivantes :





On se servira de l'outil  de la barre des taches de dessin afin de retourner le décodeur

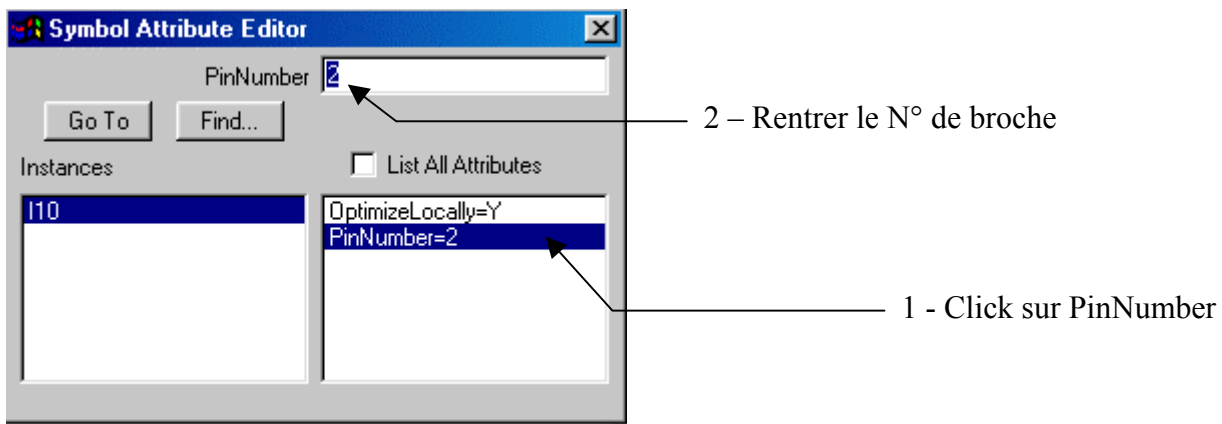
A partir de la librairie iopads.lib, placer les composants G\_OUPUT et G\_CLKBUFFER. Câbler le schéma suivant (ne pas oublier les noms et les entrées sorties) :



Les composants dans la librairie iopads, vont permettre de se connecter à une des pins du composant.

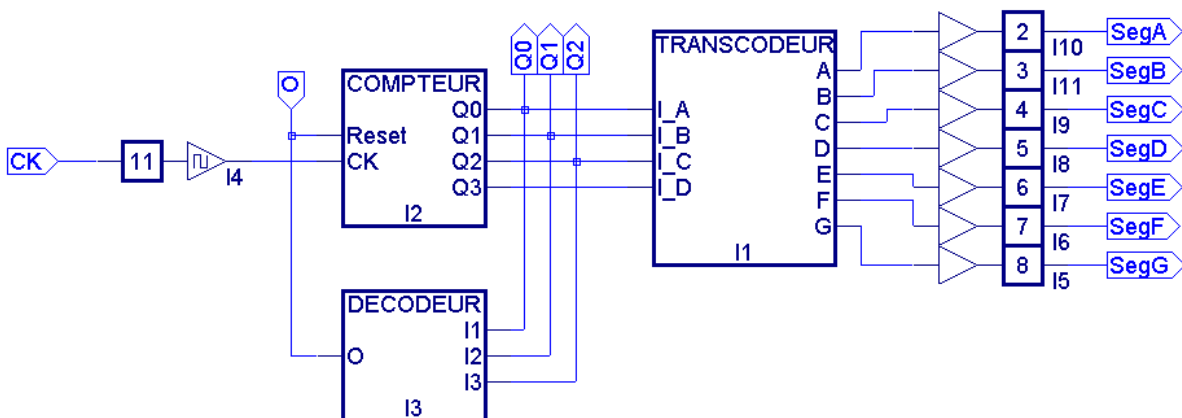
### 3.1.7 ATTRIBUTION D'UNE PIN DU COMPOSANT :

Cliquer sur l'icône Edit Symbol Attribut  de la barre des tâches de dessins. Une fenêtre apparaît, cliquer sur le premier symbole  et affecter le numéro de broche souhaité (cliquer sur PinNumber et remplacer \* par celui souhaité).

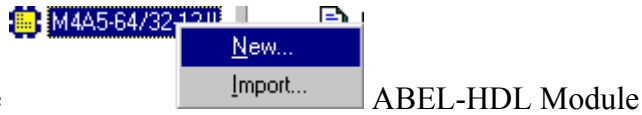


Sans taper sur la touche return cliquer sur le symbole iopads suivant et affecter le numéro de broche.

Le schéma final est celui-ci :

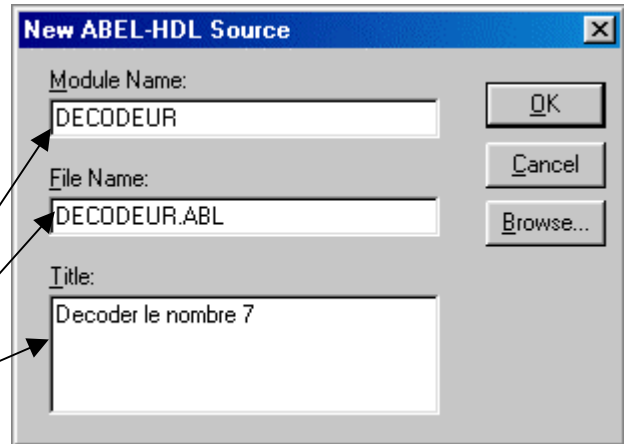


### 3.2 Associé un fichier HDL à un bloc



Sur le projet ajouter une nouvelle source

Une fenêtre s'ouvre spécifier le nom du module (identique au nom du symbole crée précédemment)  
Le fichier (extension ABL) associé à ce module et décrire sa fonction.



Nom identique au symbole

Fichier comportant le programme

Commentaire sur le programme

Un éditeur de texte s'ouvre  
Taper le programme ci-contre :

```
MODULE DECODEUR
TITLE 'Decoder le nombre 7'

Declarations
"Inputs
I1, I2, I3 PIN ISTYPE 'COM';

"Outputs
O PIN ISTYPE 'COM';

Equations

O=I1 & I2 & I3 ;

END DECODEUR
```

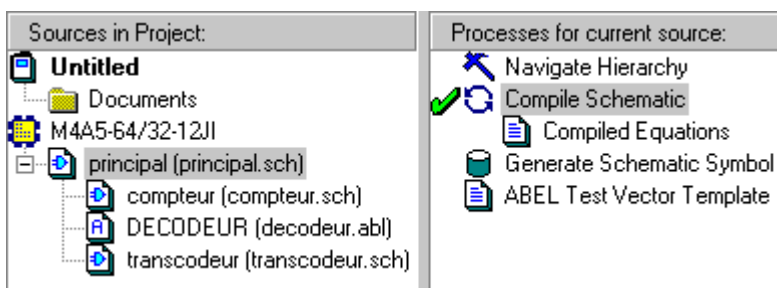
Enregistrer le programme et fermer la fenêtre.

NB : on peut aussi créer le symbole à partir du fichier abel : dans la fenêtre du projet cliquer sur le fichier abel et double cliquer sur le Generate Schematic Symbol.

### 3.3 Hiérarchie du projet

Dans le projet cliquer sur principal et compiler le schéma.

On retrouve la hiérarchie du projet avec les 3 sources filles de la source principale.



### 3.4 Simulation fonctionnelle

#### 3.4.1 AJOUT D'UNE SOURCE ABEL TEST VECTORS

Dans le projet principal ajouter une source (voir §3.1), choisir Abel Test vectors.

Donner un nom (diviseur .abv)

Dans l'éditeur de texte taper le programme suivant :

```
MODULE principal

"Déclaration Inputs
CK PIN ;

" Déclaration Outputs
O PIN ;
Q0..Q2 PIN; " Equivalent à Q0, Q1, Q2

Nb=[Q0,Q1,Q2]; " Equivalence

" Permet de décrire la simulation (entrées -> sorties)
Test_Vectors ([CK] ->[Nb, O])

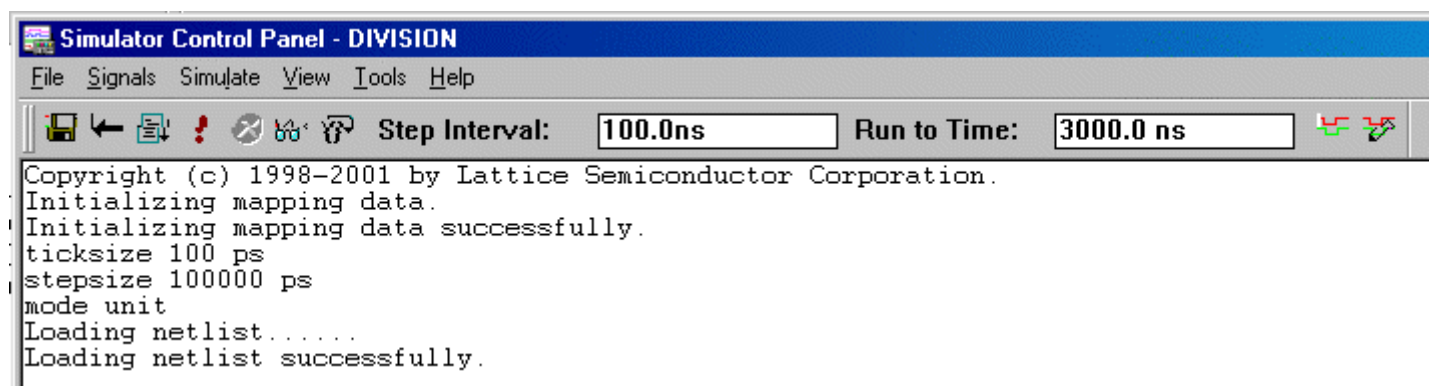
" Répète 10 fois la valeur d'entrée qui vaut une suite de 010 et calcule les 2 sorties.
" La durée d'un 0 est de 10ns.

@repeat 10 {[.c.]-> [.x., .x.];}
END
```

Enregistrer et fermer l'éditeur de texte.

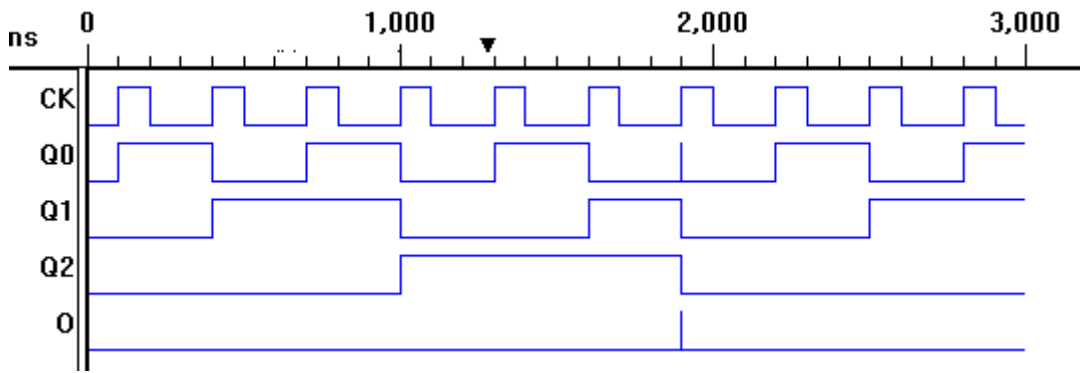
Dans la fenêtre Processes for a current source du projet, double cliquer sur compile test vector. Constaté l'absence d'erreur.

Double cliquer sur functional simulation. Après le succès de la compilation une fenêtre simulator control panel s'ouvre



Cliquer sur Run

Une fenêtre de résultats de simulation apparaît :

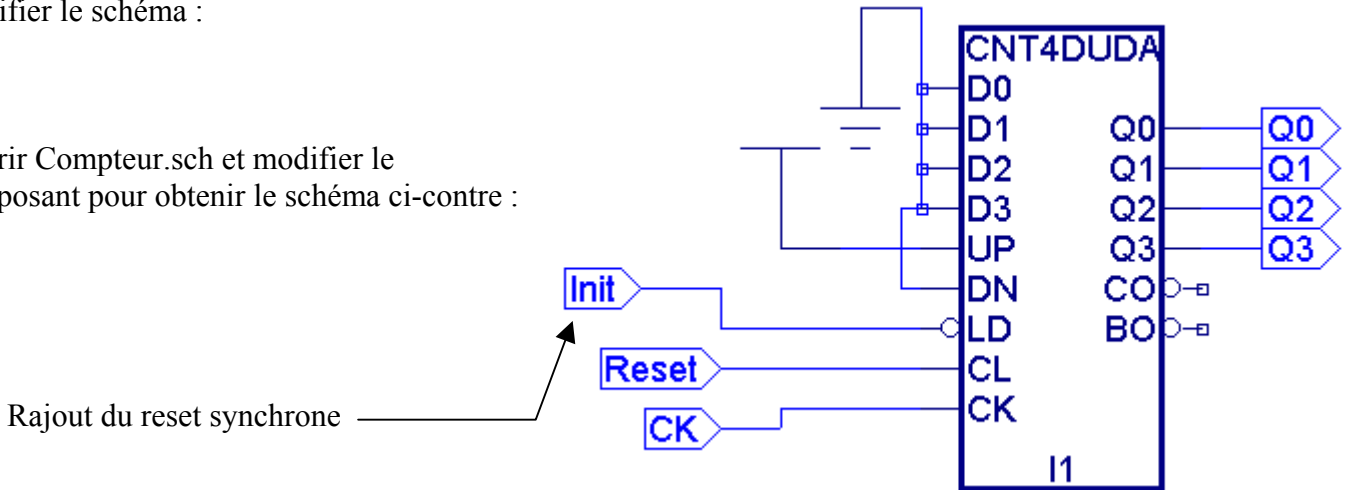


### 3.5 Simulation Temporelle

Si l'on lance la simulation temporelle on peut constater que les sorties sont dans un état indéterminé, en effet dans la réalité l'état initial est inconnu.

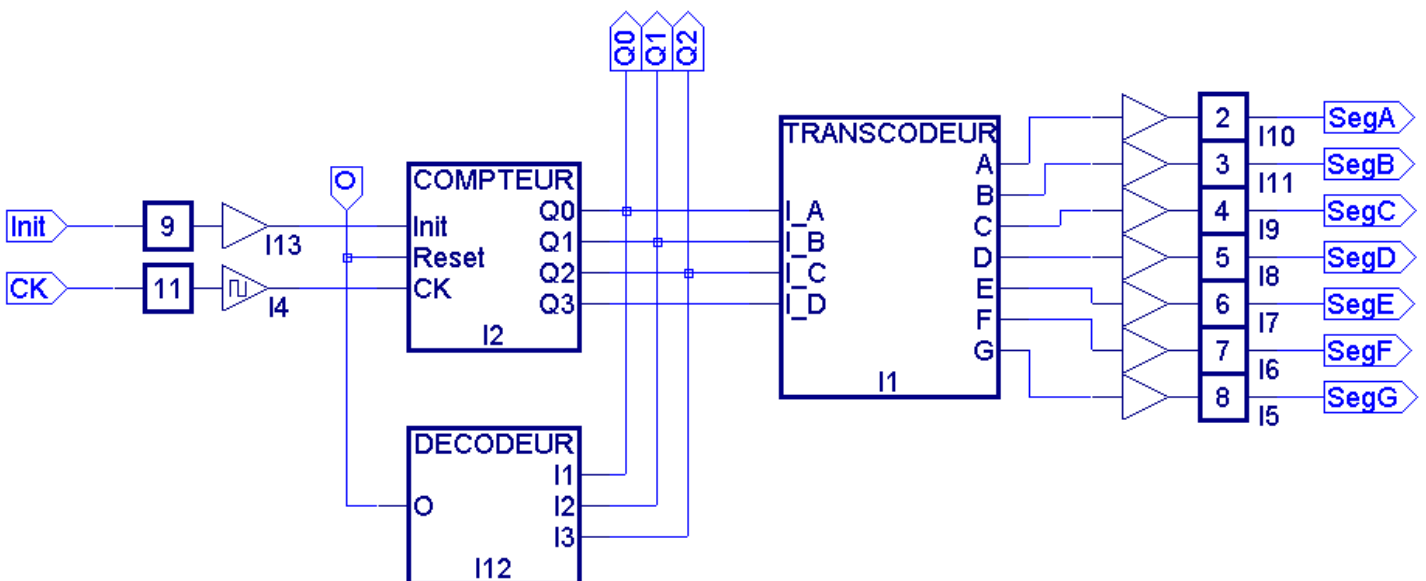
Pour montrer un résultat cohérent cela il faut modifier le schéma :

Ouvrir Compteur.sch et modifier le composant pour obtenir le schéma ci-contre :



Ecraser l'ancien symbole en créant un nouveau.

Modifier le fichier principal.sch pour obtenir le schéma suivant :



Modifier le fichier division.abv de la manière suivante :

```
MODULE principal

"Inputs
CK PIN ;
Init PIN ;


"Outputs
O PIN ;
Q0..Q2 PIN;

Nb=[Q0,Q1,Q2];

Test_Vectors ([CK,Init] ->[Nb, O])
@repeat 2 {[c.,0]-> [.x., .x.];}
@repeat 10 {[c.,1]-> [.x., .x.];}

END
```

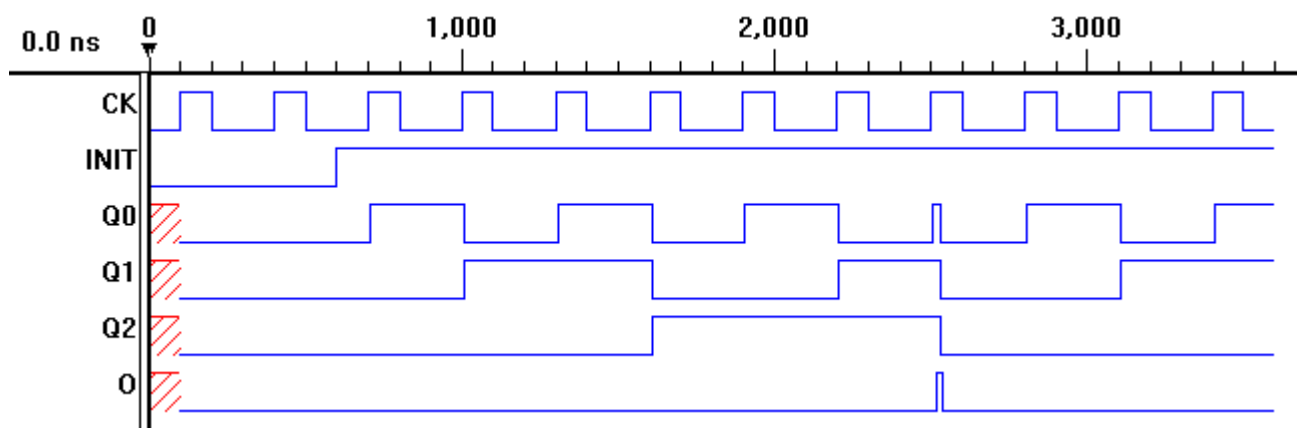
Compiler et faire une simulation temporelle

Résultat de la compilation après un run 

```
Initializing mapping data successfully.
ticksize 100 ps
stepsize 100000 ps
mode max
mode transport
Loading netlist.....

Edif netlist reader and Sdf netlist reader completed successfully.
run 3,600.0 ns
Operation RUN ends.
```

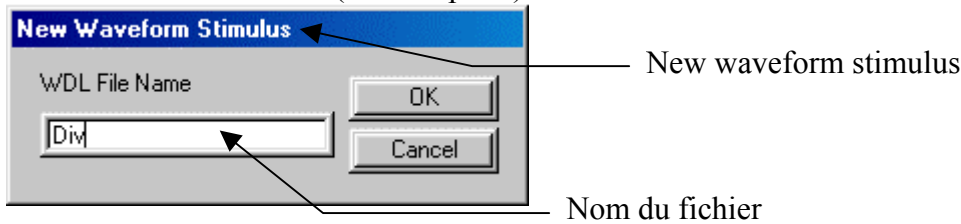
Résultat de simulation :



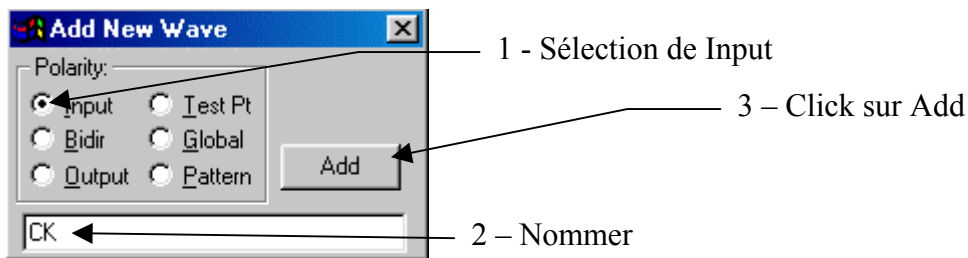
### 3.6 Ajout d'une source de stimuli graphiques pour une simulation.

On peut créer les stimuli d'entrées graphiquement à la place d'un fichier de test vector.

Créer un nouveau fichier (voir chap 3.1) waveform stimulus associé au composant M4 nom du fichier Div.



Cliquer sur l'icône New Wave . Ajouter les entrées CK et INIT (sélection de Input et Add), Ajouter Les Sorties O et Q0 (sélection de Output et Add).

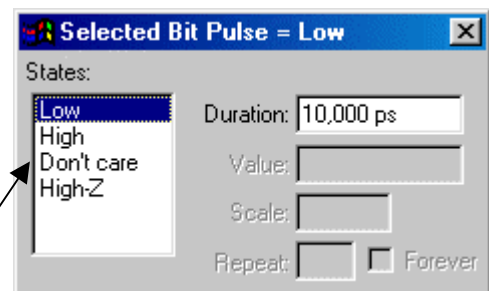


Fermer la fenêtre.

Sélectionner Edit Mode .

Cliquer sur CK I, amener la souris dans la partie chronogramme devant 10ns et sur l'alignement de CK I 0 cliquer, un niveau logique apparaît, sélectionner dans la fenêtre le niveau low (on peut modifier la durée).

Choix de l'état de l'entrée

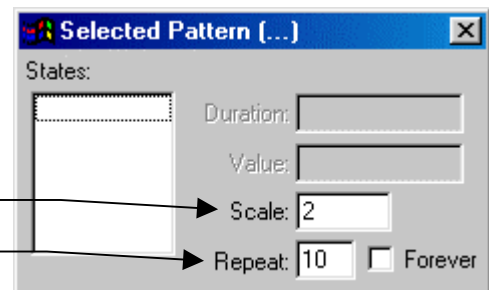


Faire de même de 20ns avec un niveau haut (automatique).

Sélectionner l'ensemble de la période (de 0 à 20ns) et dans la fenêtre rentrer 10 dans repeat (10 périodes horloge) et 2 dans Scale (double la période).

Facteur temps

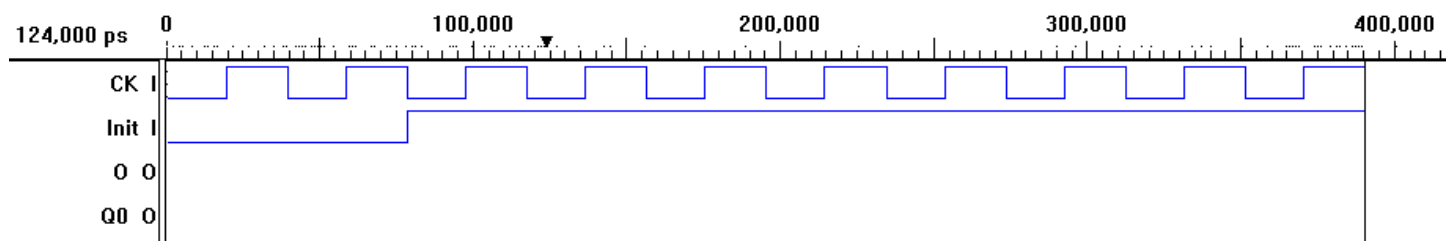
Répétition du signal



#### Stimulis init

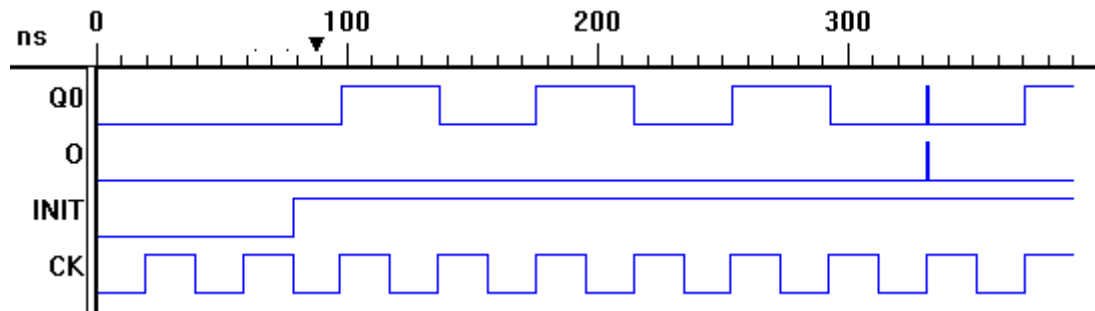
On sélectionne Init, durant les 2 premières périodes de l'horloge on force à 0, le reste du temps à 1.

Le résultat graphique est le suivant :

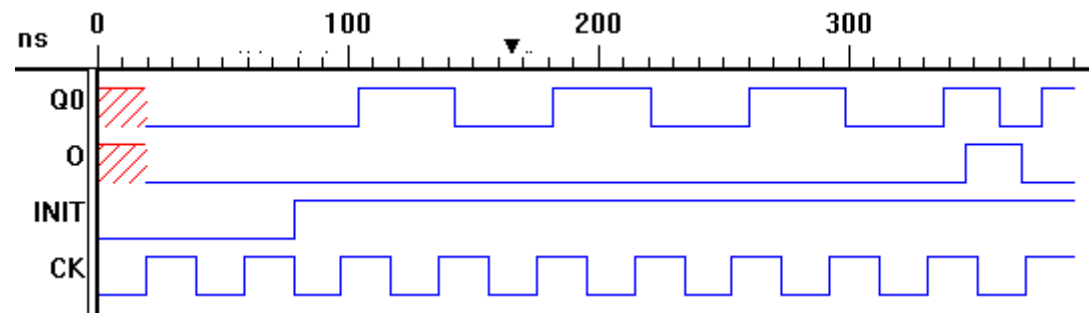


Enregistrer et fermer la fenêtre de stimulis.

Dans le projet cliquer sur le fichier div.wdl, puis double click sur functional simulation ensuite sur RUN dans la fenêtre de dialogue le résultat est le suivant :



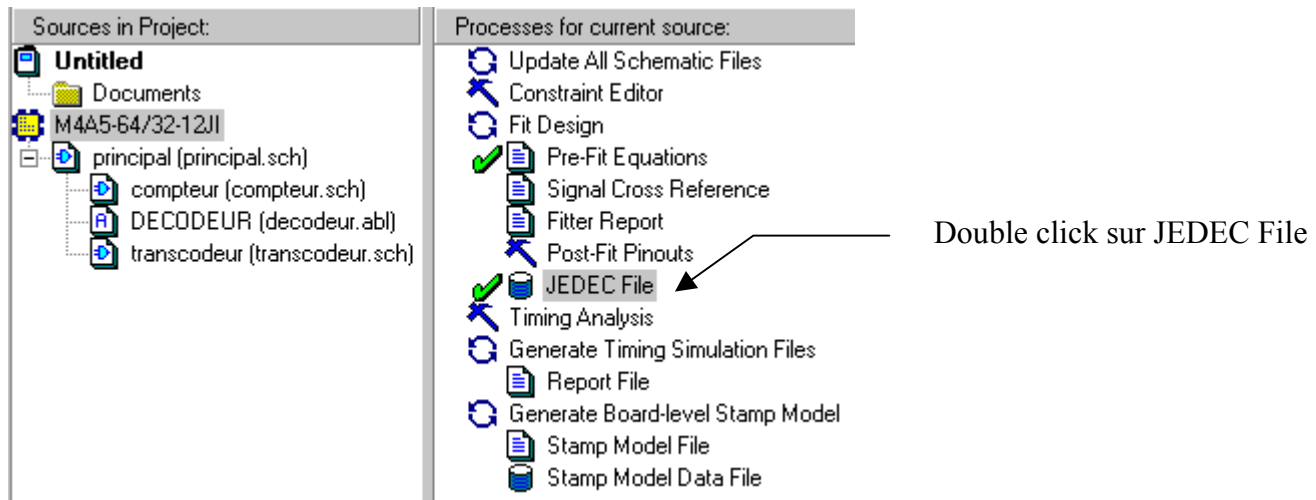
Même chose avec timing simulation :



## 4 PROGRAMMATION D'UN COMPOSANT

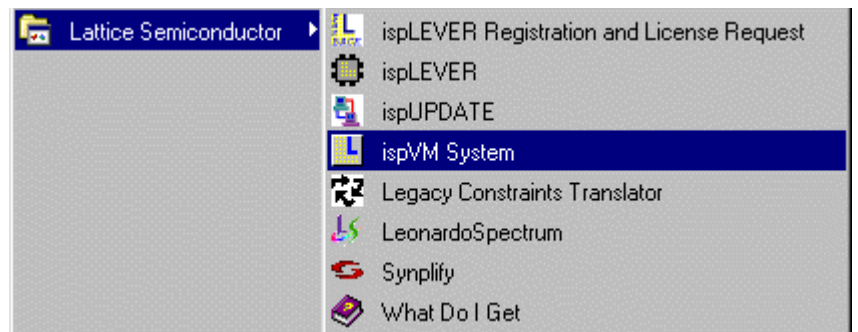
### 4.1 Génération du fichier JEDEC

Dans le projet cliquer sur le composant MA4A5-64-32-12JI et dans la fenêtre processus double cliquer sur JEDEC File et répondre YES aux différentes fenêtres.



### 4.2 Choix d'un fichier JEDEC et d'un circuit à programmer

Demarrer ispVM System



Créer un nouveau fichier :



Une fenêtre chain configuration s'ouvre.

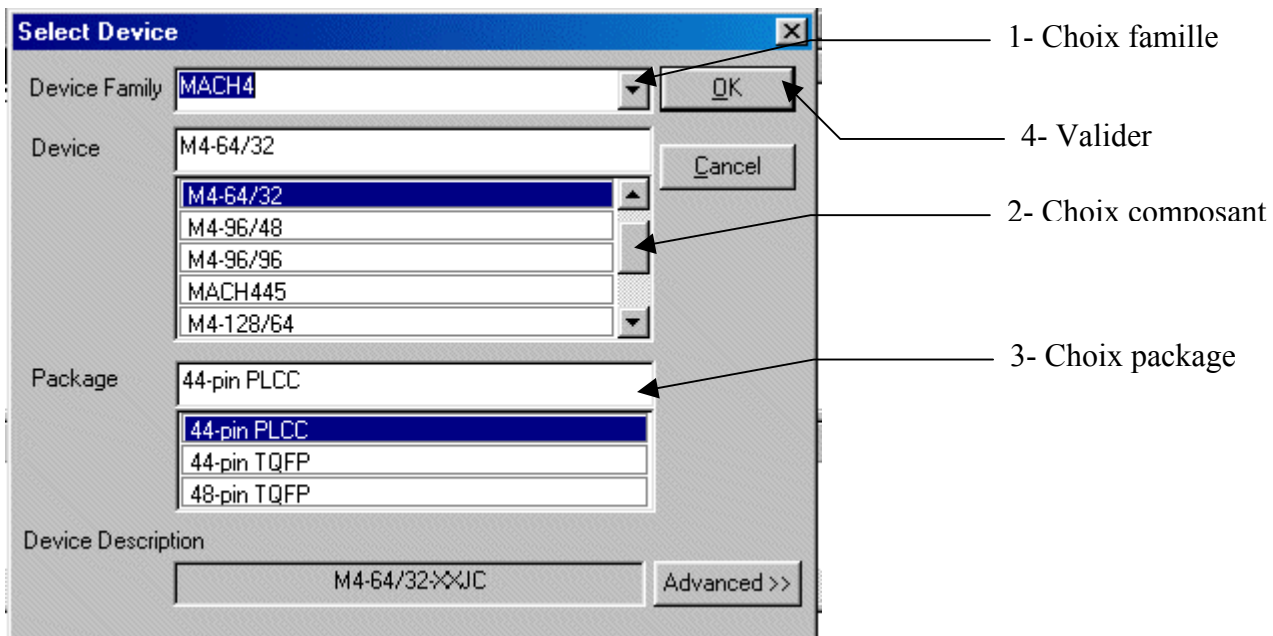
Cliquer sur add device



Une fenêtre à compléter s'ouvre

Dans le champ device selec mentionner le nom du composant à programmer : M4-64/32

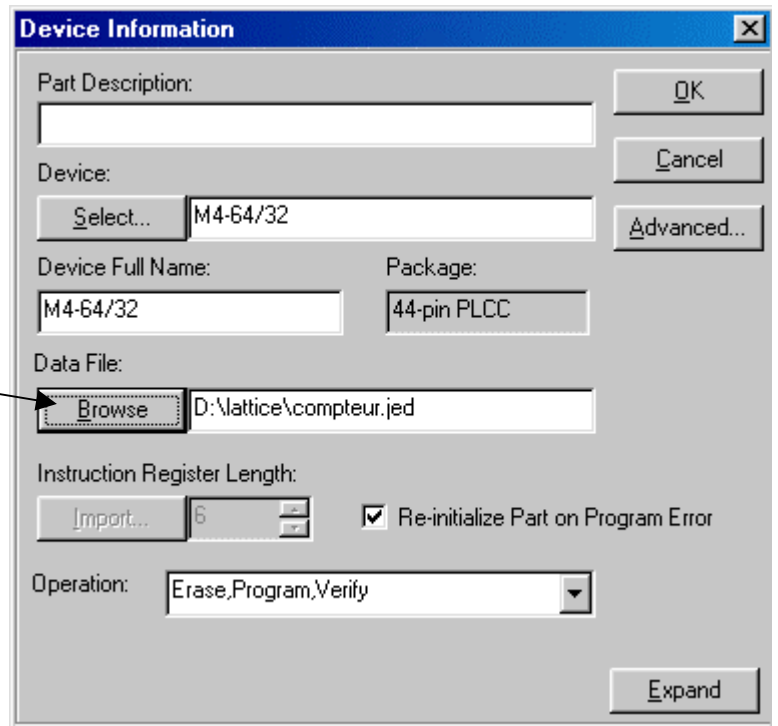
Cliquer sur select et choisir la famille, le composant et le package exact.



Renseigner le champ data file en choisissant le fichier JEDEC ayant le nom du projet.

Choisir le composant adéquat et le fichier de programmation .jed

Choix du fichier JEDEC



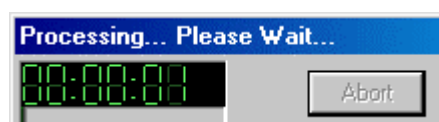
On trouve le résumé des choix effectués :

Index	Device List	FileName/IR-Length	Operation	Status
1	M4-64/32	D:\lattice\compteur.jed	Erase,Program,Verify	N/A

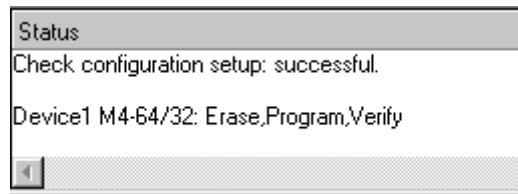
Cliquer sur GO



Une fenêtre d'attente apparaît :



A sa disparition le résultat de programmation apparaît dans la fenêtre du bas d'ispVM System



Le circuit est programmé

FIN