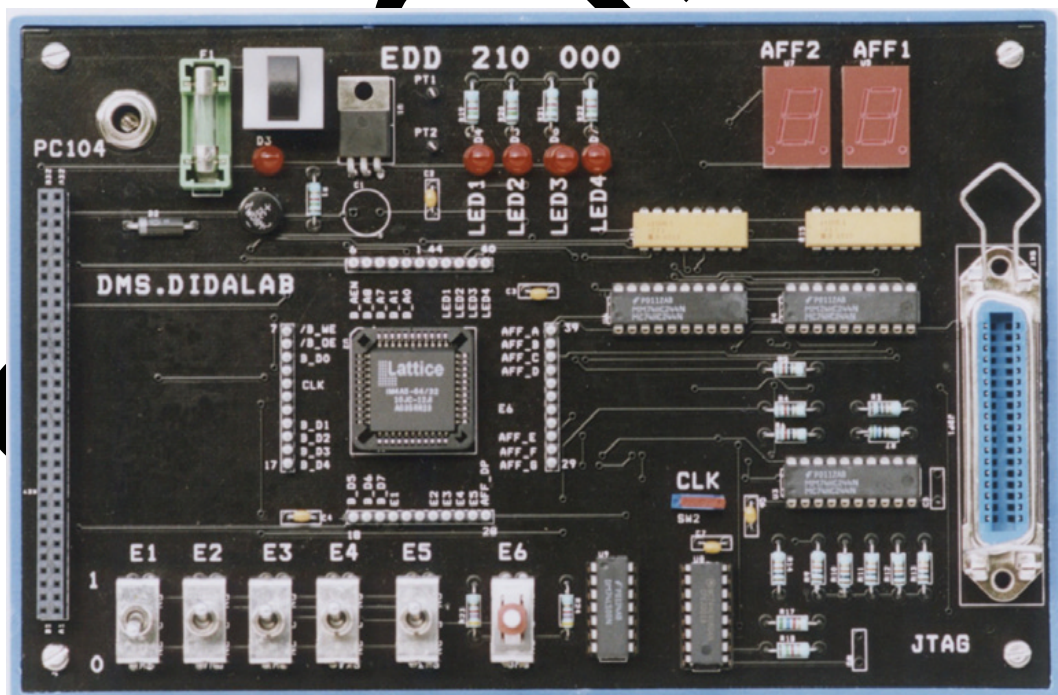


MANUEL ELEVE

TRAVAUX PRATIQUES

LOGIQUE PROGRAMMEE SUR
EPLD



SPECIMEN

SOMMAIRE

TP 0	: Mise en œuvre de l'EDD200 du Pack d'étude de l'EPLD, réalisation de la fonction inverseur sur 4 bits.	5
TP 1	LOGIQUE COMBINATOIRE SIMPLE	26
TP 2	MULTIPLEXEUR	27
TP 3	BASCULE RS ELEMENTAIRE	30
TP 4	COMPT. / DECOMPT. BCD et PREDIV. PROGRAMMABLE	32

SPECIMEN

SPECIMEN

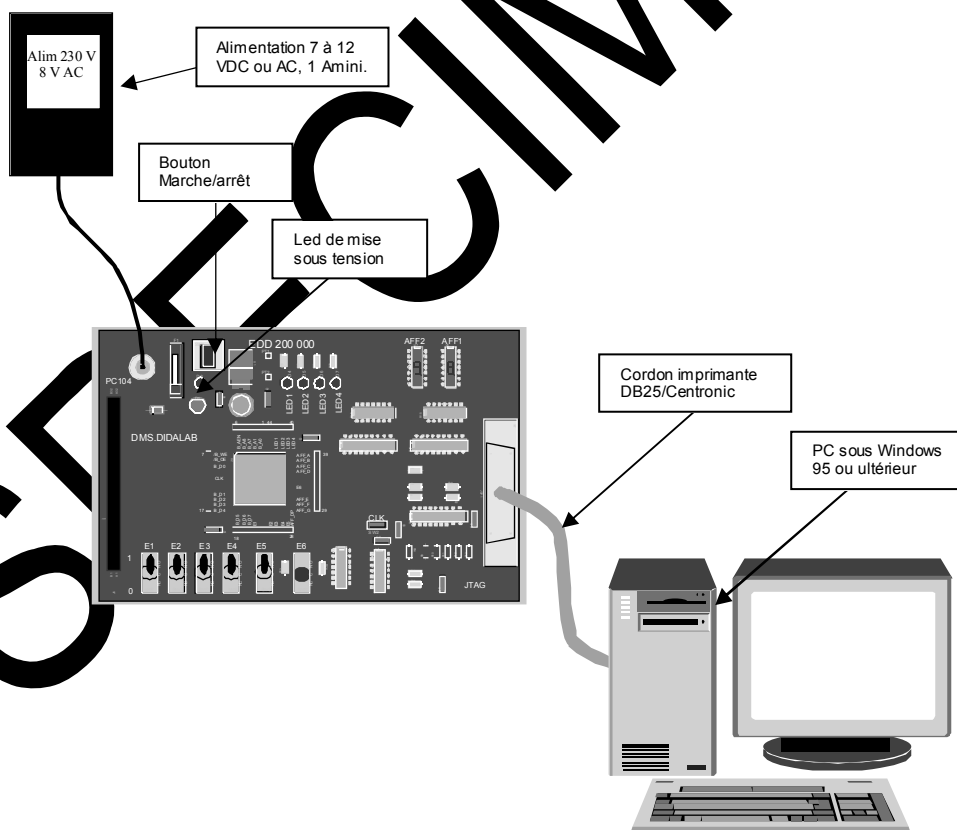
TP 0 : Mise en œuvre de l'EDD200 du Pack d'étude de l'EPLD, réalisation de la fonction inverseur sur 4 bits.

0.1 Avertissement

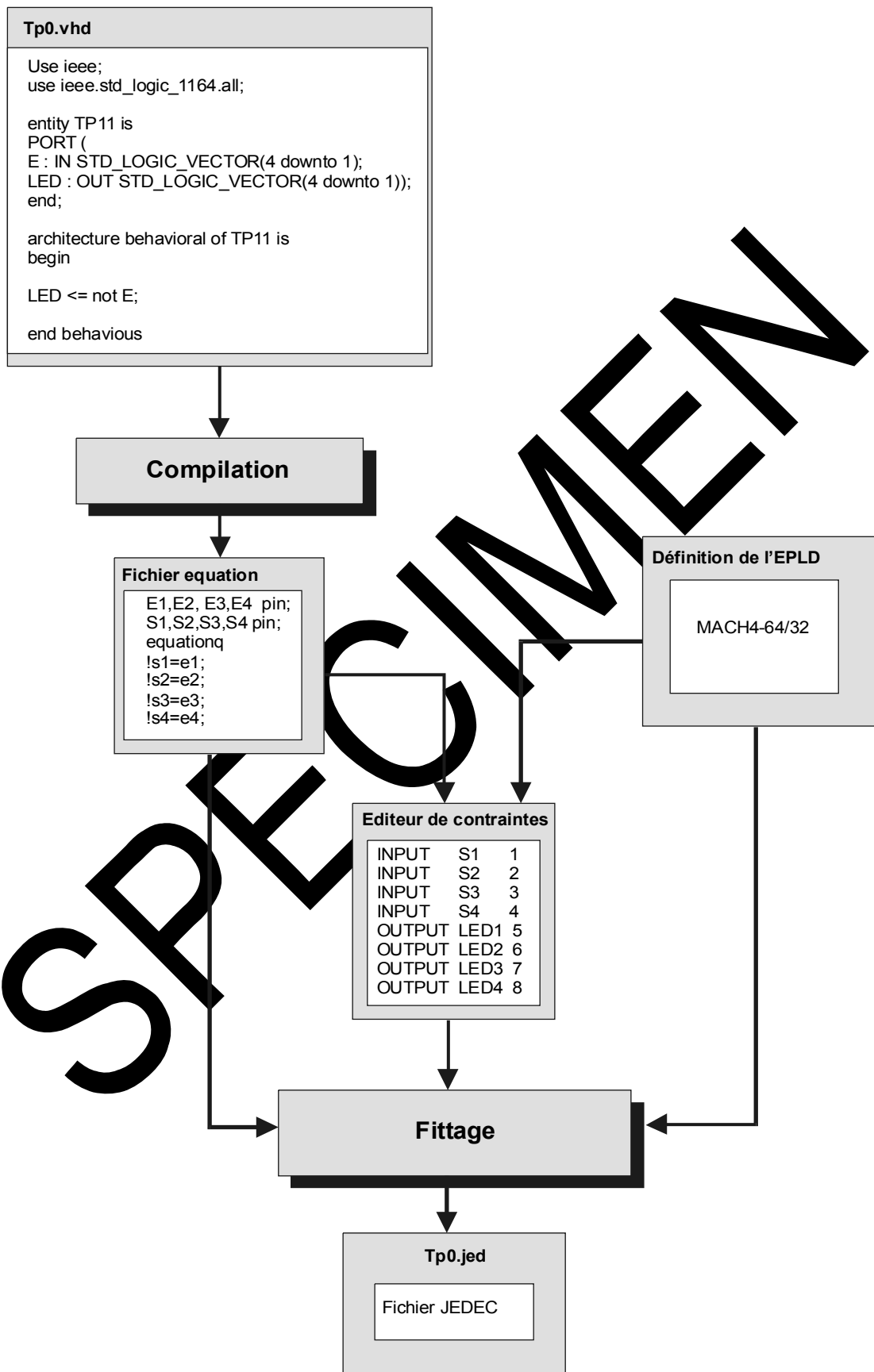
Nota : La fiche de TP décrite ci-après n'a aucune ambition pédagogique, elle a simplement pour but d'aider l'utilisateur à la prise en main de l'ensemble logiciel et matériel EDD200 pack d'étude de l'EPLD. Elle est composée d'étapes successives très détaillées de la mise en œuvre du matériel et logiciel lors de la première utilisation.

0.2 Installation matériel.

- 0.2.1 Relier la carte EDD 200 000 au PC de développement VHD par le cordon imprimante,
- 0.2.2 Connecter le boîtier alimentation sur la carte EDD 200 000, (7 à 12 V AC ou DC),
- 0.2.3 Appuyer sur le bouton Marche Arrêt de la carte EDD 200 000, la lampe témoin rouge doit s'allumer.



0.3 Présentation du déroulement d'une phase complète de développement VHDL.



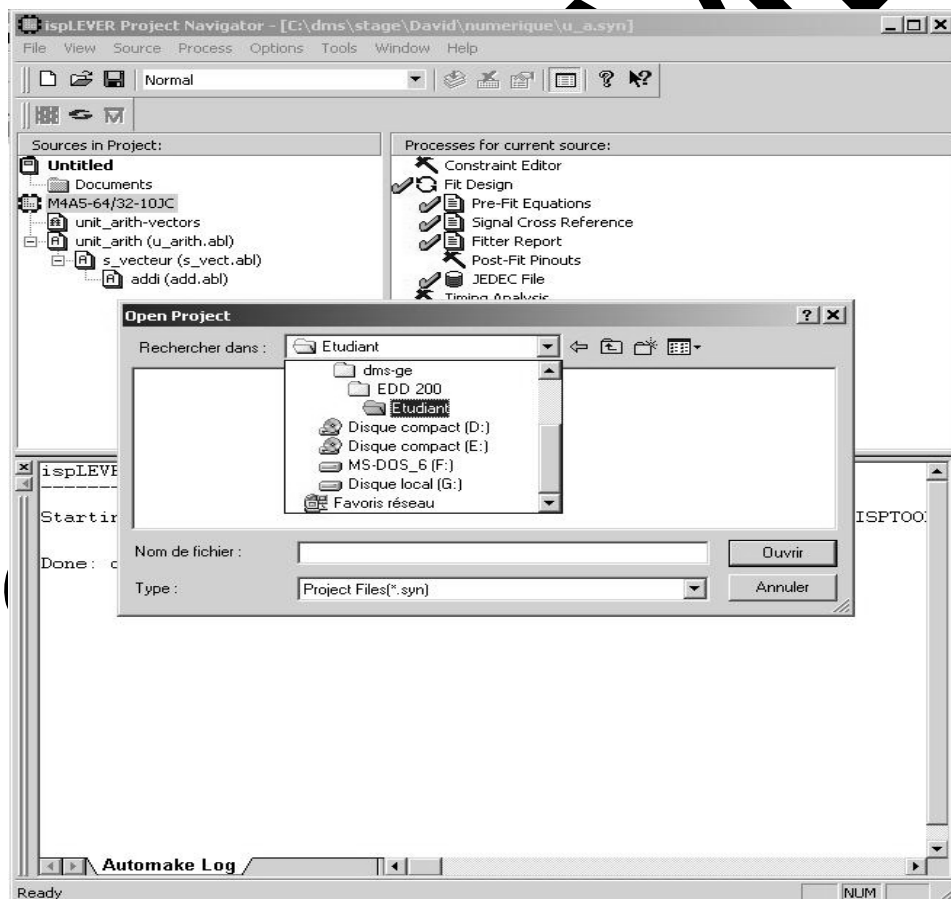
0.4 Démarrage du logiciel.

- 0.4.1 Cliquer sur « **démarrer** »,
- 0.4.2 Cliquer sur « **programme** »,
- 0.4.3 Cliquer sur « **Lattice Semiconductor** »,
- 0.4.4 Cliquer sur « **ispLEVER** »,



0.5 Ouverture du fichier exemple : Tp0.vhd.

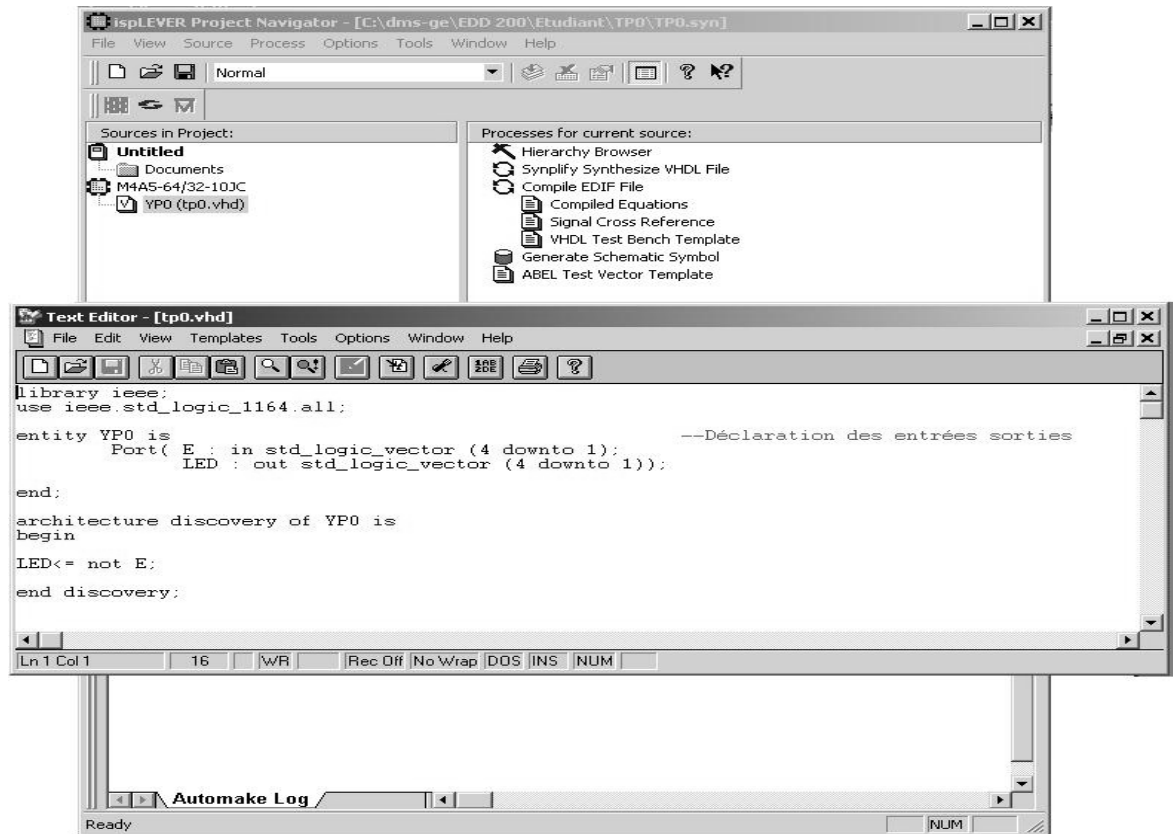
- 0.5.1 Cliquer sur « **File** »,
- 0.5.2 Cliquer sur « **Open Project** »,
- 0.5.3 Cliquer sur « **C :** », (par la fenêtre de sélection du chemin),
- 0.5.4 Cliquer sur **dms-ge**,
- 0.5.5 Cliquer sur **Edd200**,
- 0.5.6 Cliquer sur **Etudiant**,
- 0.5.7 Cliquer sur **tp0**,
- 0.5.8 Cliquer sur **Ouvrir**,



0.6 Edition du fichier exemple : Tp0,

0.6.1 Dans la fenêtre « Sources in Project :) » :

Double cliquer sur «  TP0 (tp0.vhd) ».



Nous pouvons observer les différentes zones du fichier tp0.vhd ci-dessus :

- **Library ieee** : inclusion de la bibliothèque standard,
- **Use ieee.std_logic_1164.all** : utilisation des ports d'entrées sorties standards,
- **Entity** : définition des ports d'entrées sorties globaux,
- **Architecture** : portion de description du comportement.

0.7 Paramétrage du compilateur :

Double cliquer sur la fonction composant « figure ci dessous », pour ouvrir la fenêtre de paramétrage.

Vérifier, ou introduire les valeurs comme indiqué ci-dessous :

Paramètres à valider :

Family = ispMACH4A5,

Device = M4A5-64/32,

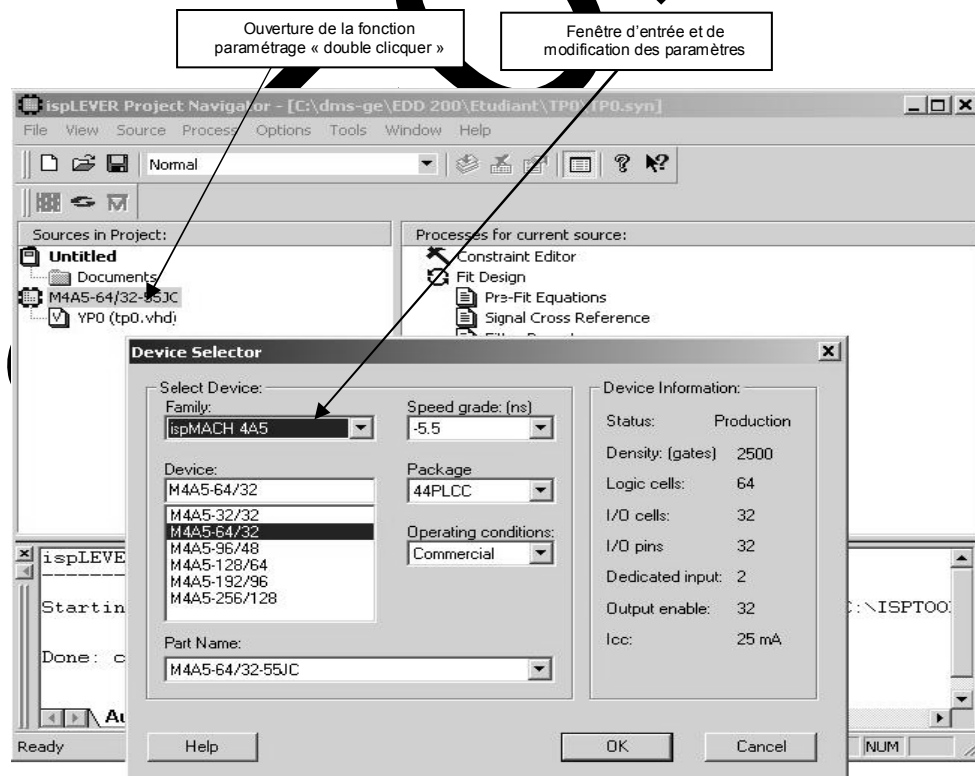
Speed grade (ns) = -5.5

Package = 44PLCC,

Operating condition = commercial,

Part name = M4A5-64/32-55JC.

Cliquer sur “OK”

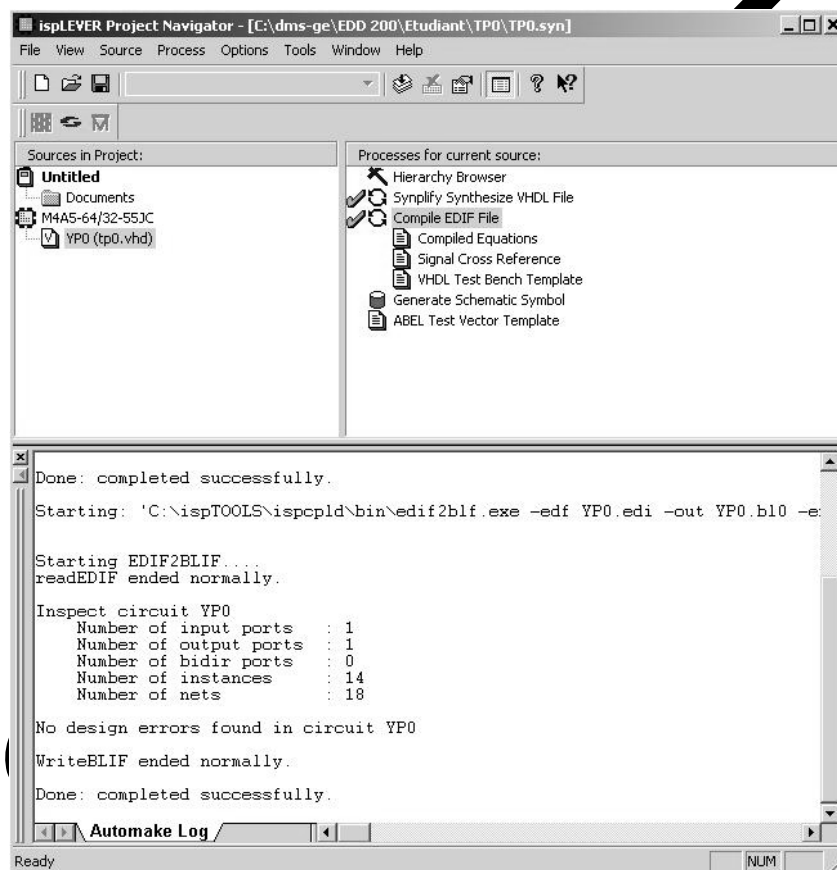


0.8 Compilation du fichier VHDL.

Cliquer sur « **V** TP0 (tp0.vhd) »

Pour démarrer double cliquer sur : « **Compile EDIF File** »,

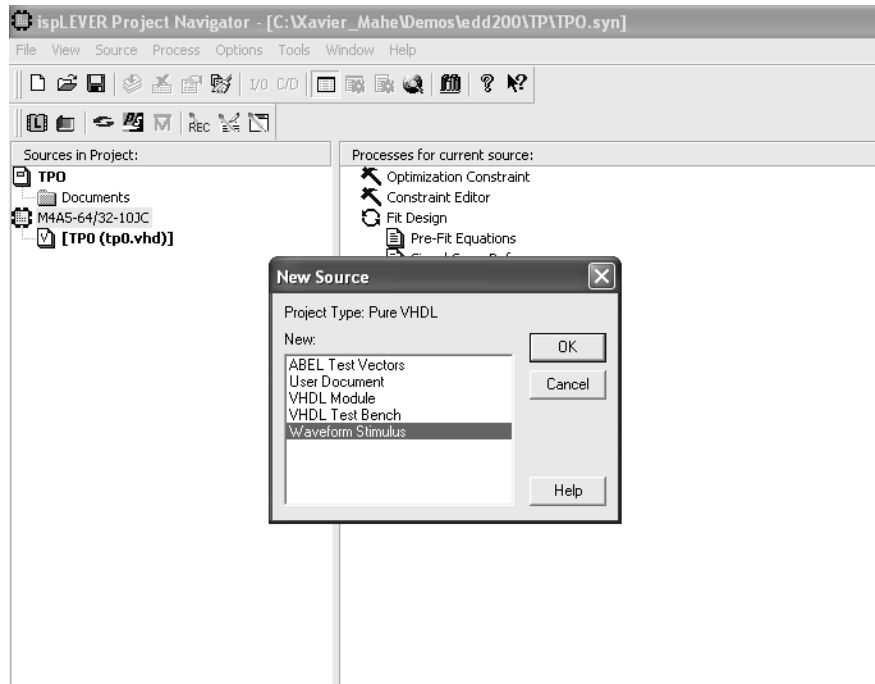
- Si aucune modification n'a été faite dans le fichier TP0, le compilateur répondra « **Process (Compile edit file) is up to date** », les symboles sont verts et autorisent un téléchargement dans le circuit.
- Si une modification a été apportée dans le fichier d'exemple Tp0, le compilateur édite les nouvelles équations, et annonce des avertissements à l'utilisateur indiquant que les alimentations ne sont plus attribuées et doivent être rétablies. Cela signifie que l'utilisateur doit procéder au fittage du circuit, affectation des entrées sorties.



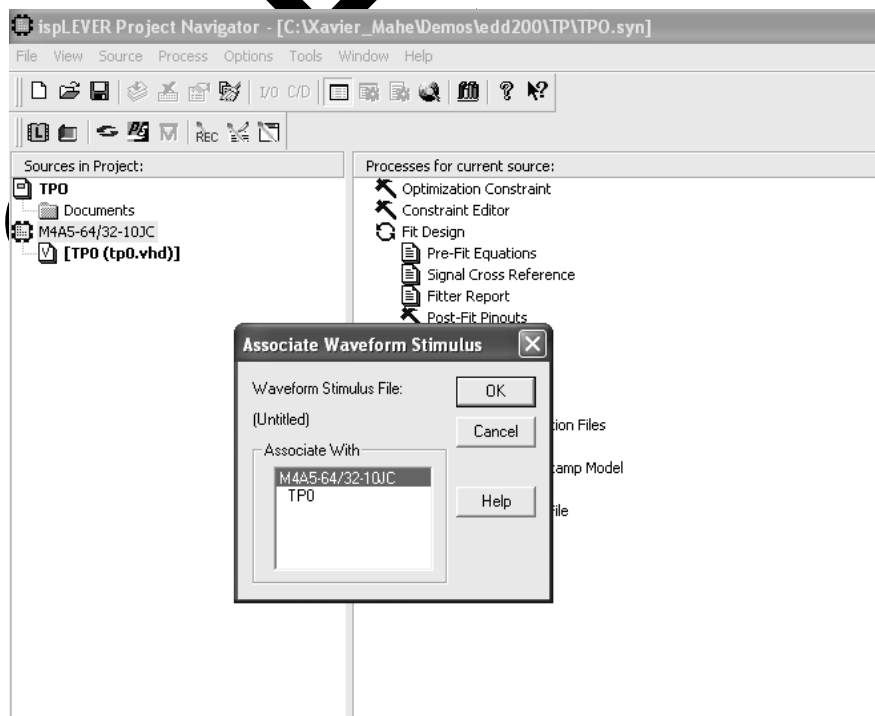
0.9 Simulation de la description VHDL

0.9.1 Création du fichier de simulation

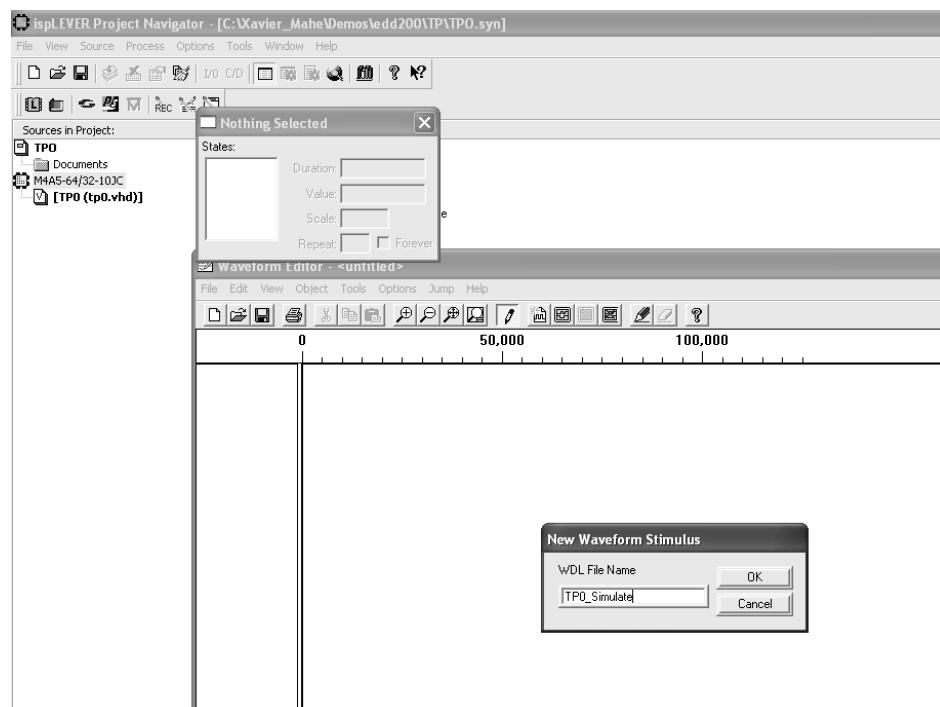
Cliquer droit sur « **M4A5-64/32-10JC** »,
Sélectionner « **new** »
Sélectionner « **waveform stimulus** »



Sélectionner « **M4A5-64/32-10JC** »

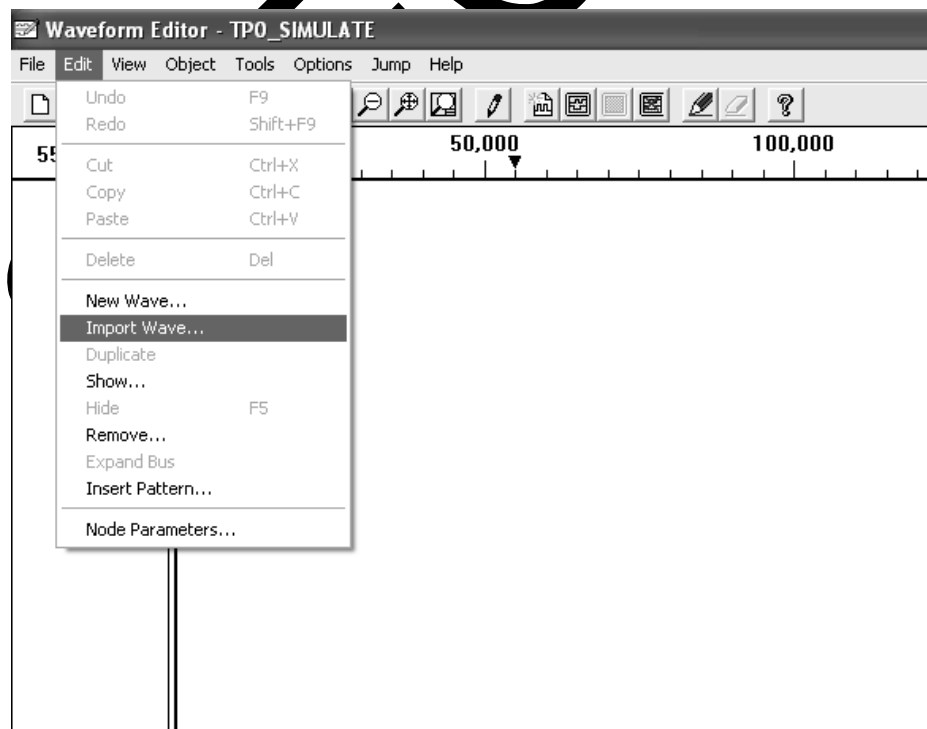


Donner un nom au fichier de simulation (par exemple : « TP0_Simulate »)

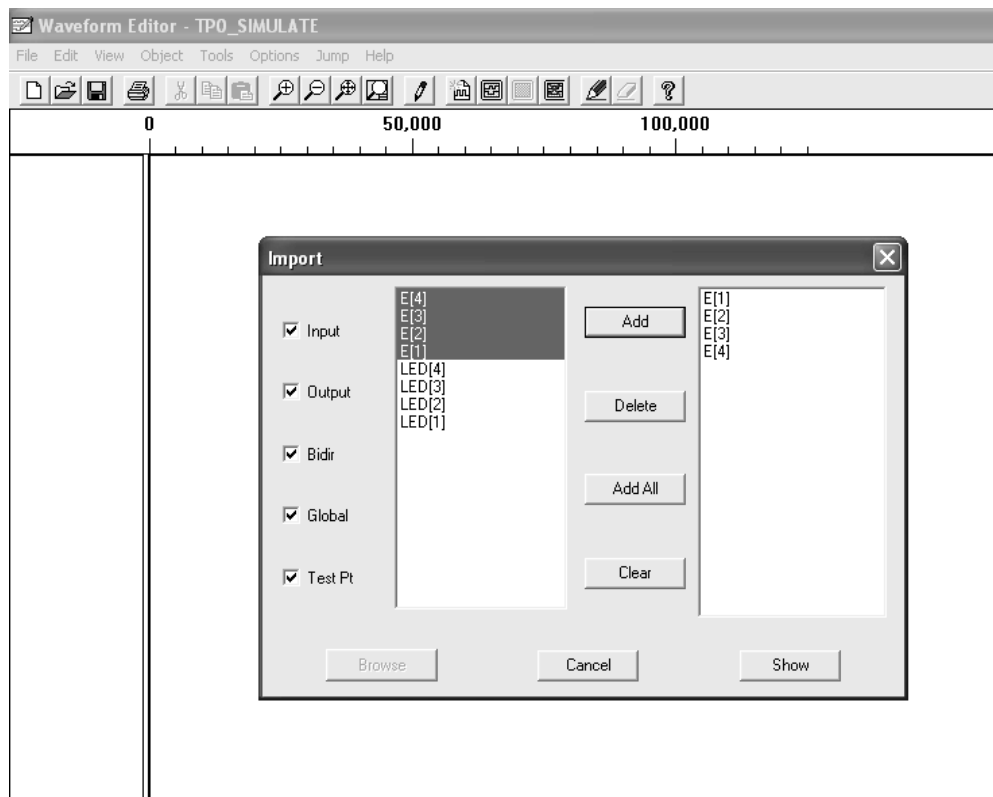


0.9.2 Définition des entrées du système

Sélectionner « **import wave** » dans le menu « **Edit** » du Waveform Editor



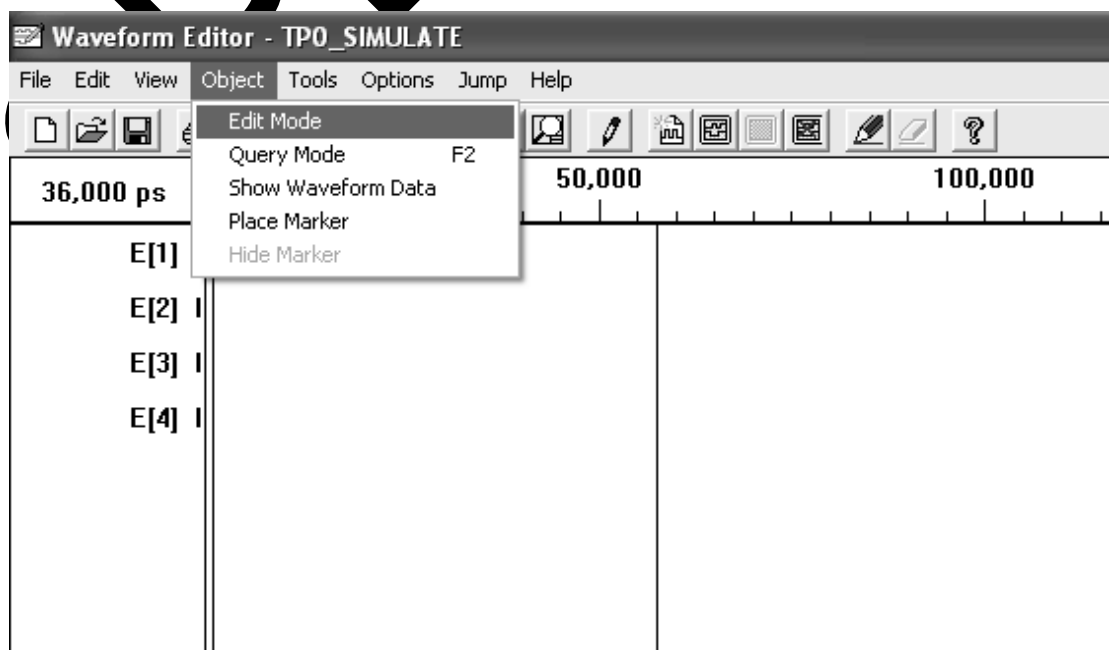
Sélectionner les entrées E(1) à E(4) et cliquer sur « **Add** » afin de les insérer dans la fenêtre d'édition des Stimuli.



Cliquer sur « **Show** » pour fermer la fenêtre « **Import** »

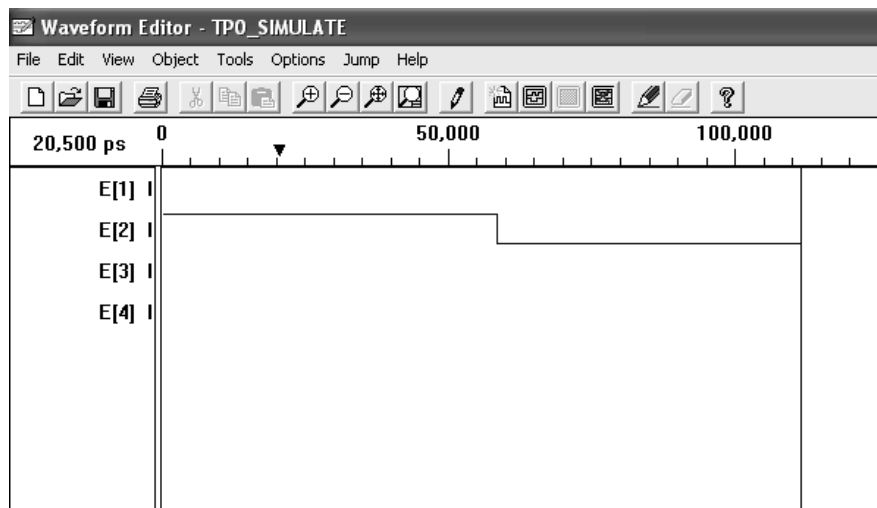
0.9.3 Création des chronogrammes

Dans le menu « **Object** », sélectionner « **Edit Mode** »



Exemple de création sur E(2) :

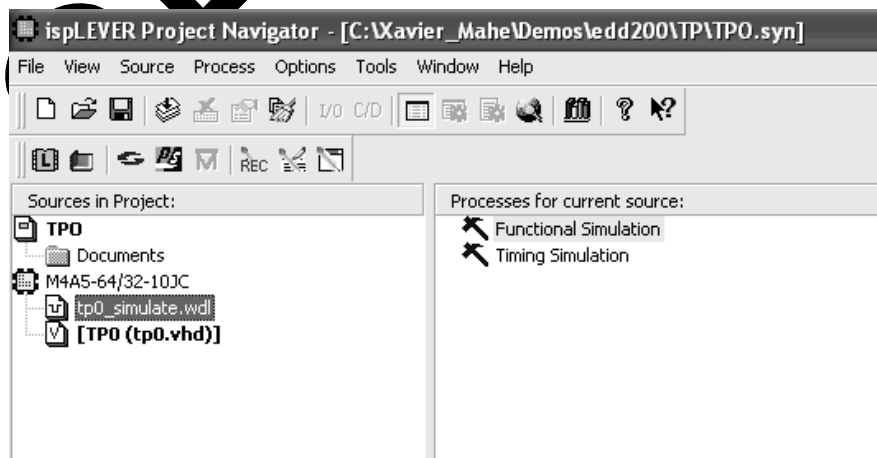
- Cliquer sur la zone graphique à l'horizontale de E(2), faire un glisser horizontal, le chronogramme se positionne par défaut à l'état haut,
- Cliquer ensuite à droite de cette même zone E(2) et faire un glisser, le chronogramme se positionne par défaut à l'état bas,
- etc



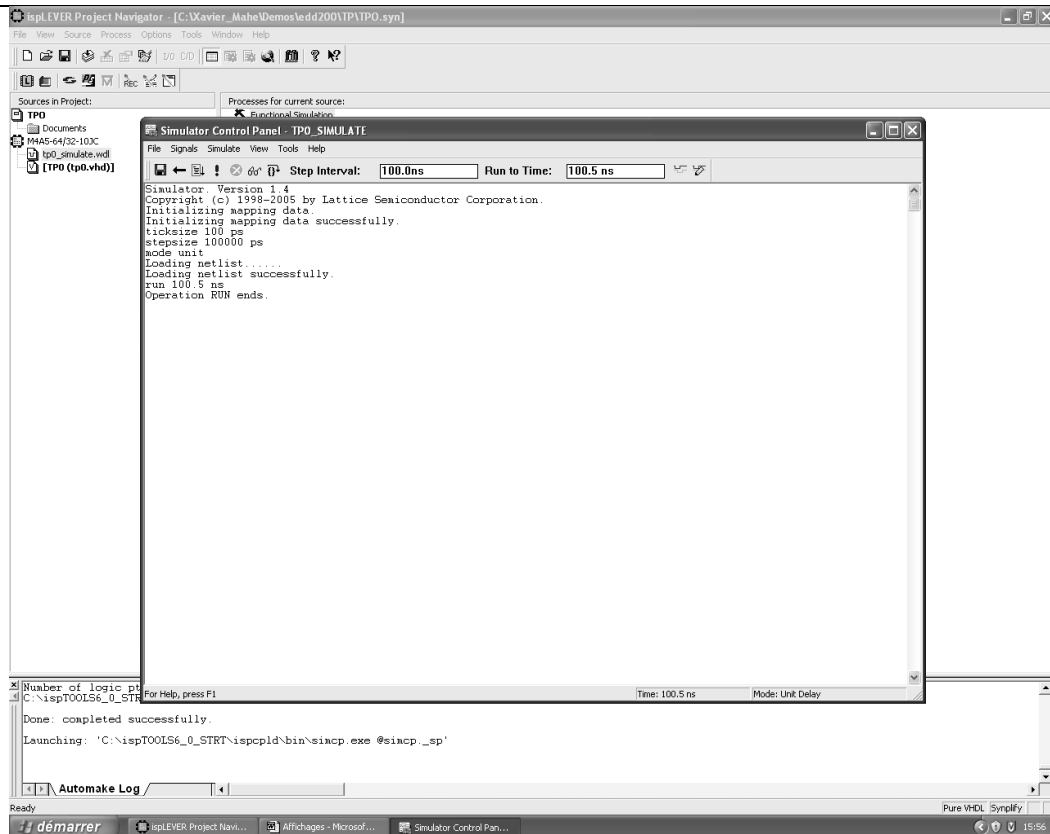
Procéder de même pour chacun des signaux d'entrée
Enregistrer le fichier TPO_Simulate
Sortir du Waveform Editor

0.9.4 Lancement de la simulation

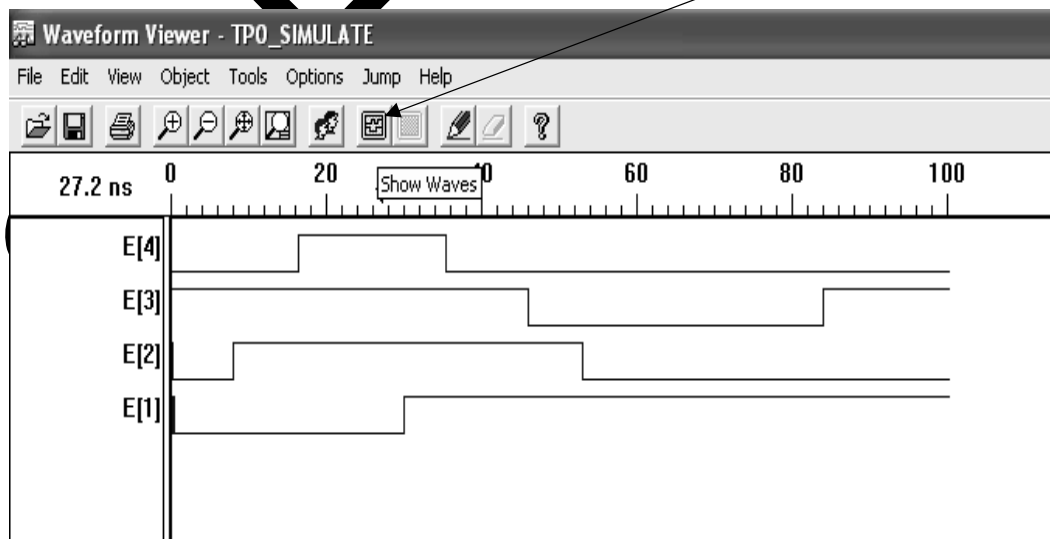
Retourner dans ISpLEVER Project Navigator
Se positionner sur TPO_Simulate.wdl



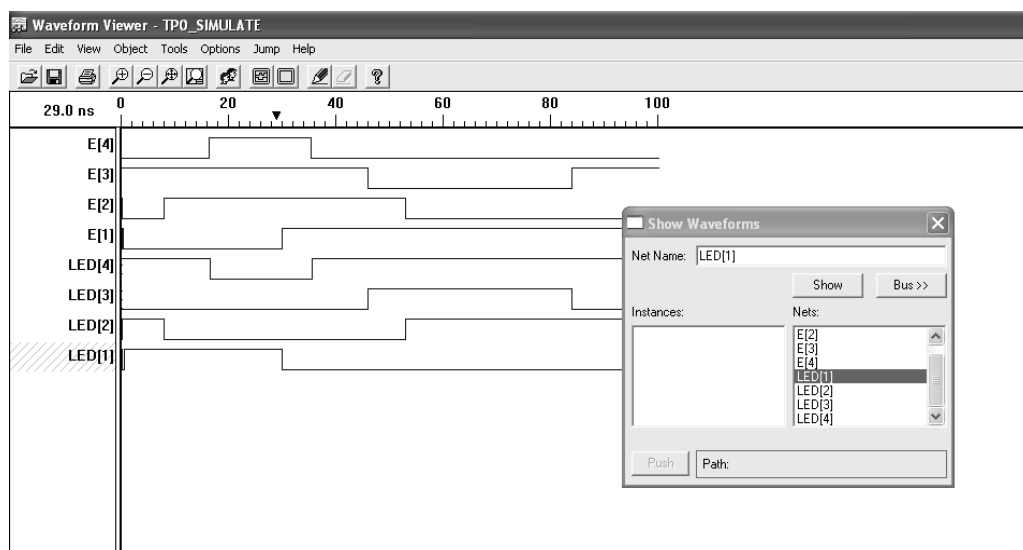
Double-cliquer sur « **Fonctionnal Simulation** »
La fenêtre « Simulator Control Panel » s'ouvre



Dans le menu « **Simulate** », cliquer sur « **Run** »
 La fenêtre « **Waveform Viewer** » s'ouvre
 Pour afficher le résultat de la simulation, cliquer sur l'icône « **Show waves** »



Double-cliquer sur chacune des sorties LED(1) à LED(4) à visualiser



On affiche ainsi le résultat de la simulation

0.10 Programmation du brochage du circuit.

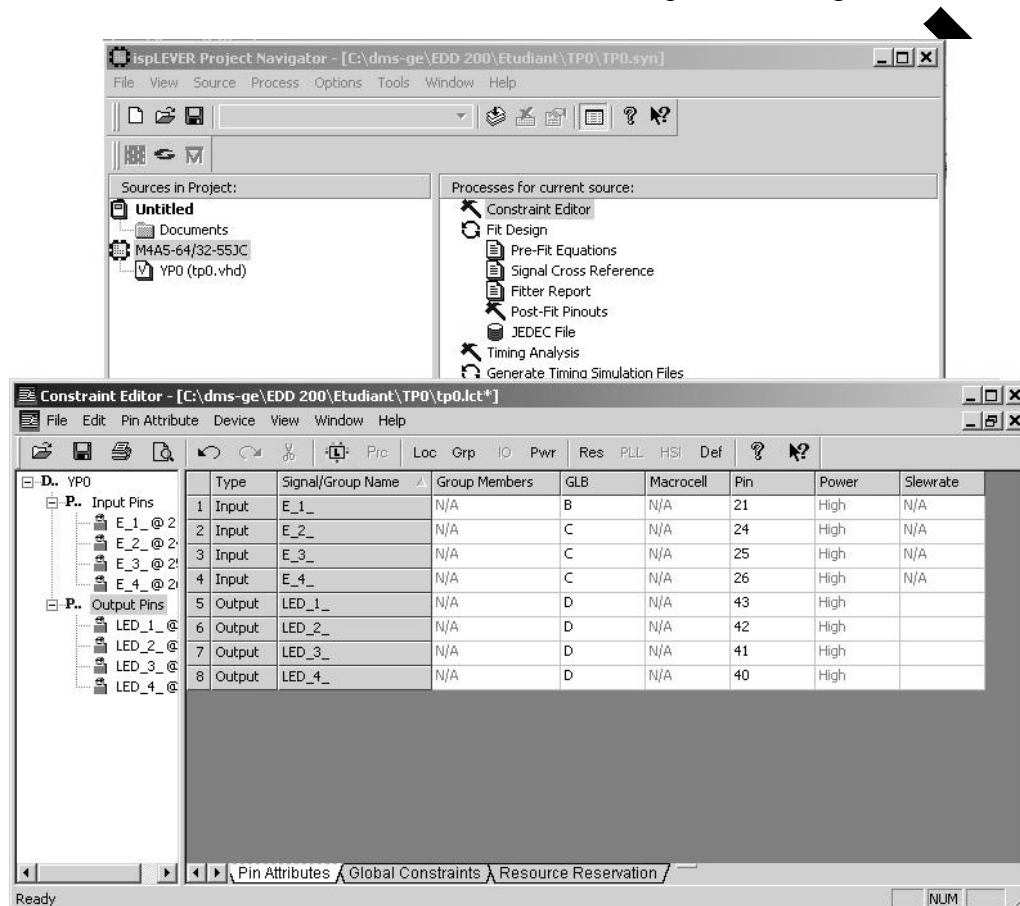
0.10.1 Ouverture du fichier

Cliquer sur « **M4A5-64/32-55JC** »,

Double cliquer sur « **Constraint Editor** », une nouvelle fenêtre s'ouvre

Double cliquer sur TP0, puis sur Input Pins et Output Pins

Vérifier les fonctions d'entrées/sorties comme indiqué dans la figure ci-dessous.



0.10.2 Création des brochages, (nota : il est important de bien respecter les brochages indiqués sur la figure ci-dessus).

Pour simuler une création de brochage, Cliquer sur « **loc** »,

Supprimer au préalable l'entrée E_1, (cliquer sur la ligne puis Delete),

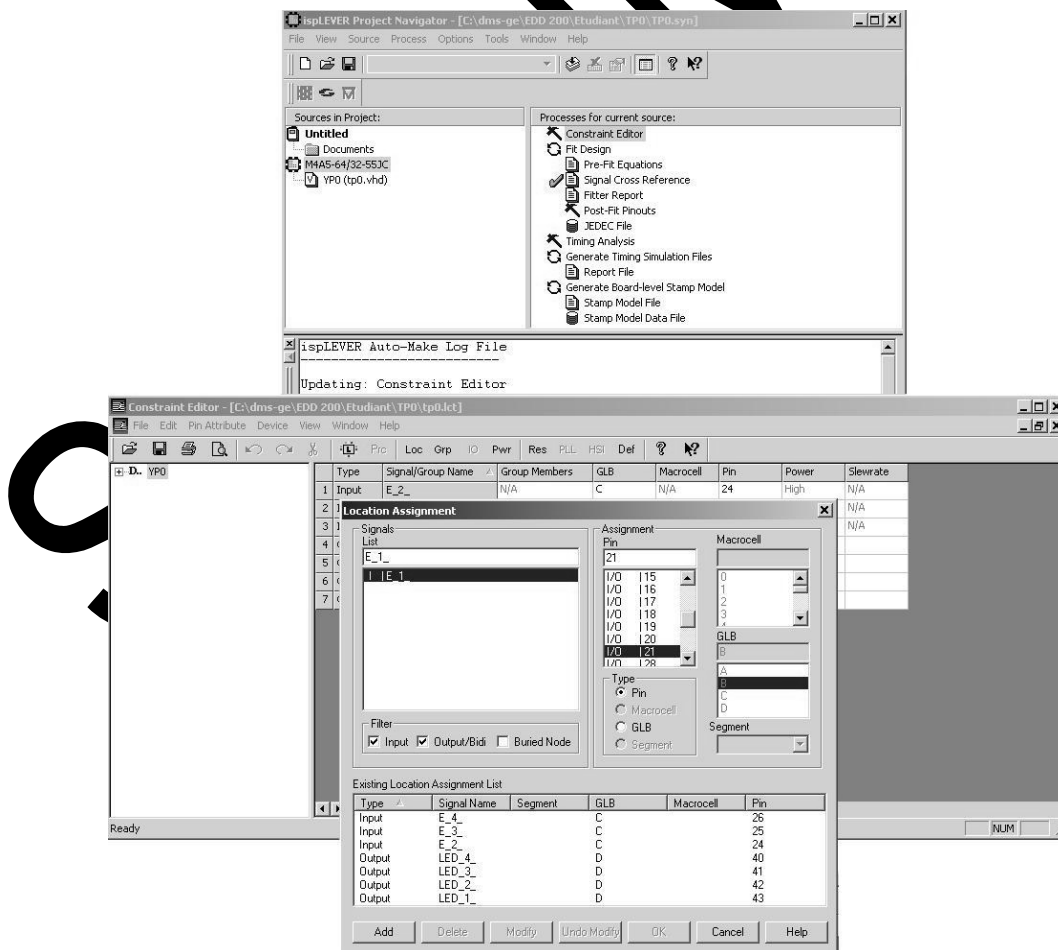
La ligne manquante (broche 21 apparaît dans la fenêtre « **Signals** »).

Pour créer le brochage :

- Cliquer sur « **I/O 21** », dans la fenêtre « **Assignement Pin** »
- Cliquer sur « **I I E_1_** », dans la fenêtre « **Signals List** pour le valider dans la ligne supérieur comme ci-dessous.
- Déterminer son comportement, dans la cas présent « **input** »,

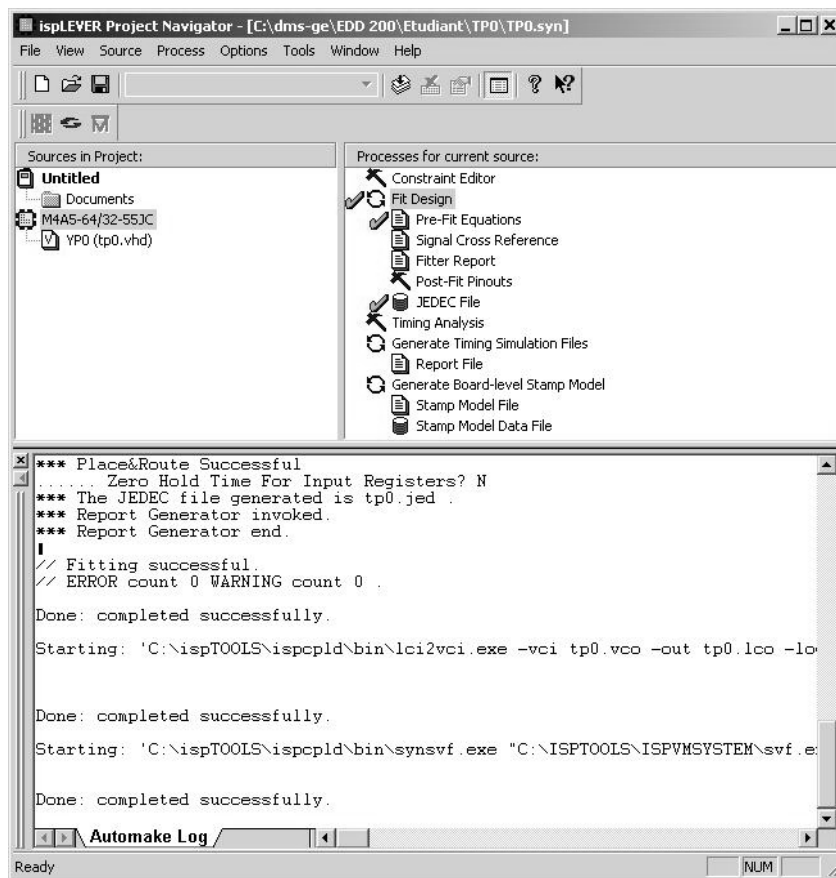
Nota : une sortie serait en réalité une « **output/Bidi** »

Valider et quitter le programme. (fermer le logiciel « **Isi Design EXPERT** »)



0.11 FITTAGE

0.11.1 Double Cliquer sur « Fit Design »



0.12 Téléchargement du fichier JEDEC dans le circuit EPLD à travers le JTAG :

0.12.1 Préparation du téléchargement.

Par le menu Démarrer : lancer le programme « **ispVM System** ».

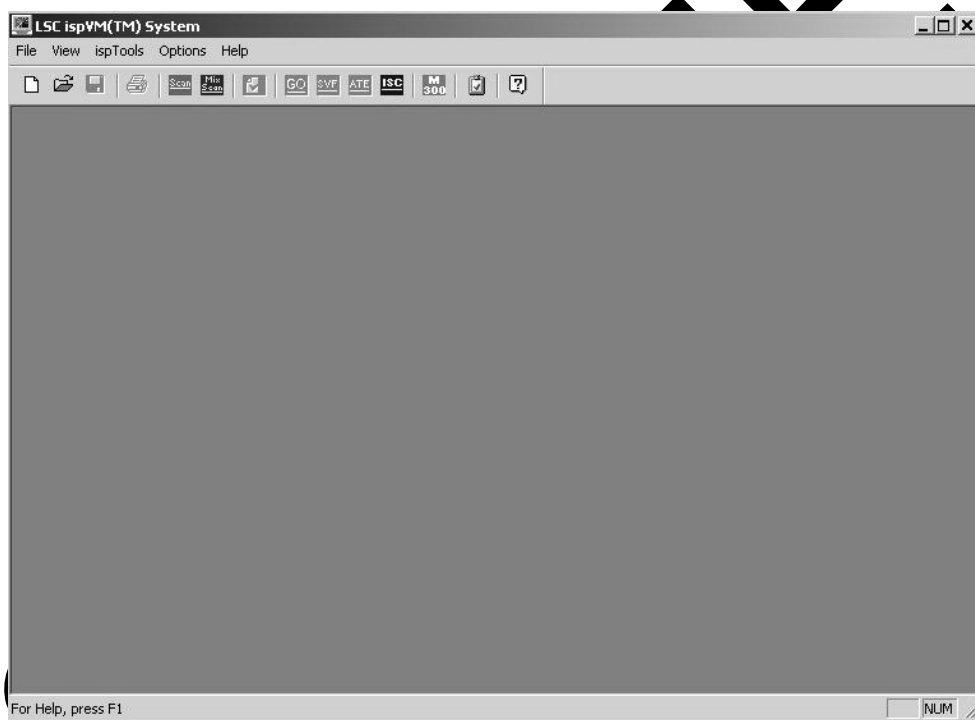
0.12.2 Cliquer sur « **démarrer** ».

0.12.3 Cliquer sur « **programme** ».

0.12.4 Cliquer sur « **Lattice Semiconductor** ».

0.12.5 Cliquer sur « **ispVM System** ».

La fenêtre suivante apparaît



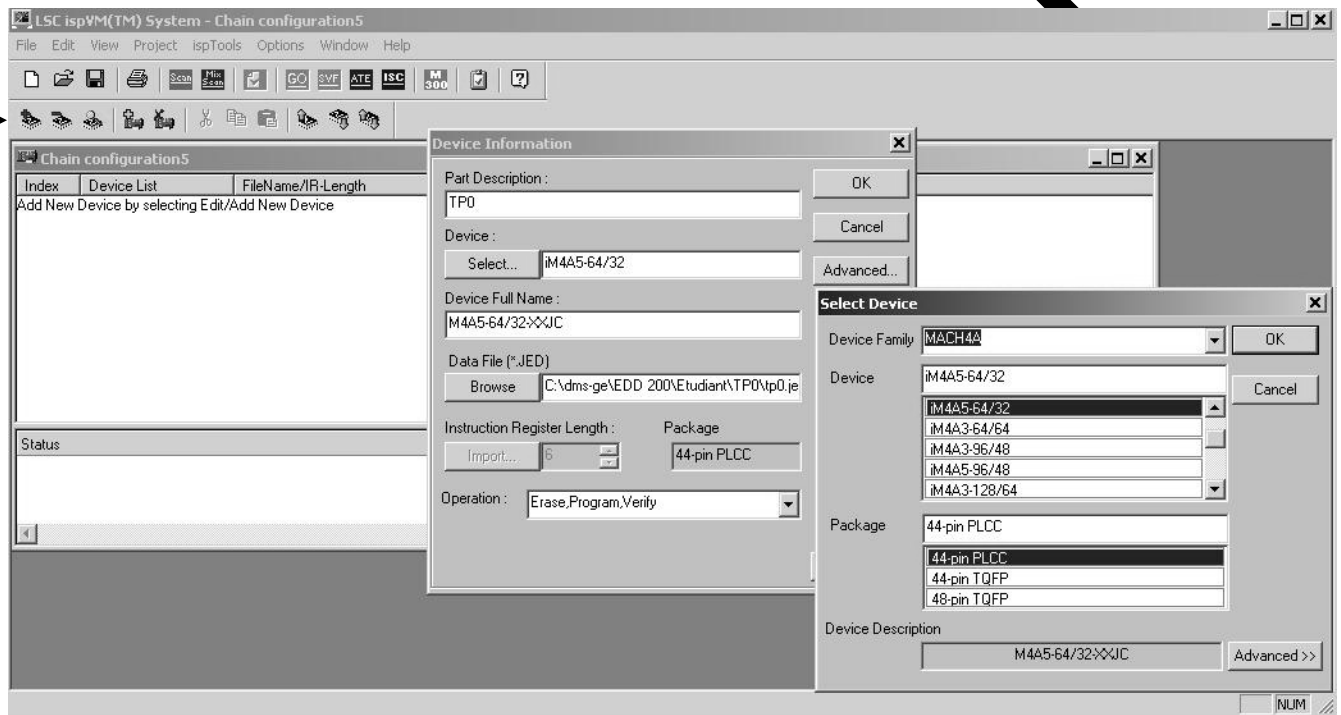
Cliquer sur « **File** », puis sur « **New** »

Appuyer sur **Add Device** (touche insert).

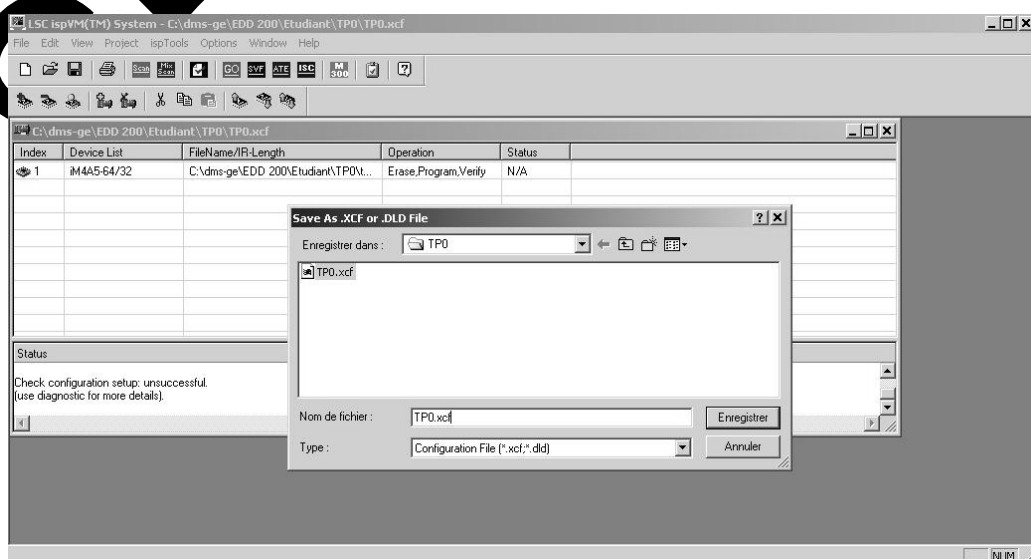
Sélectionner le composant comme au préalable en appuyant sur select

Puis dans browse sélectionner le fichier jed se trouvant dans le répertoire **C:\dms-ge\EDD200\Etudiant\TP0**.

Appuyer sur OK



Puis sauvegarder la configuration



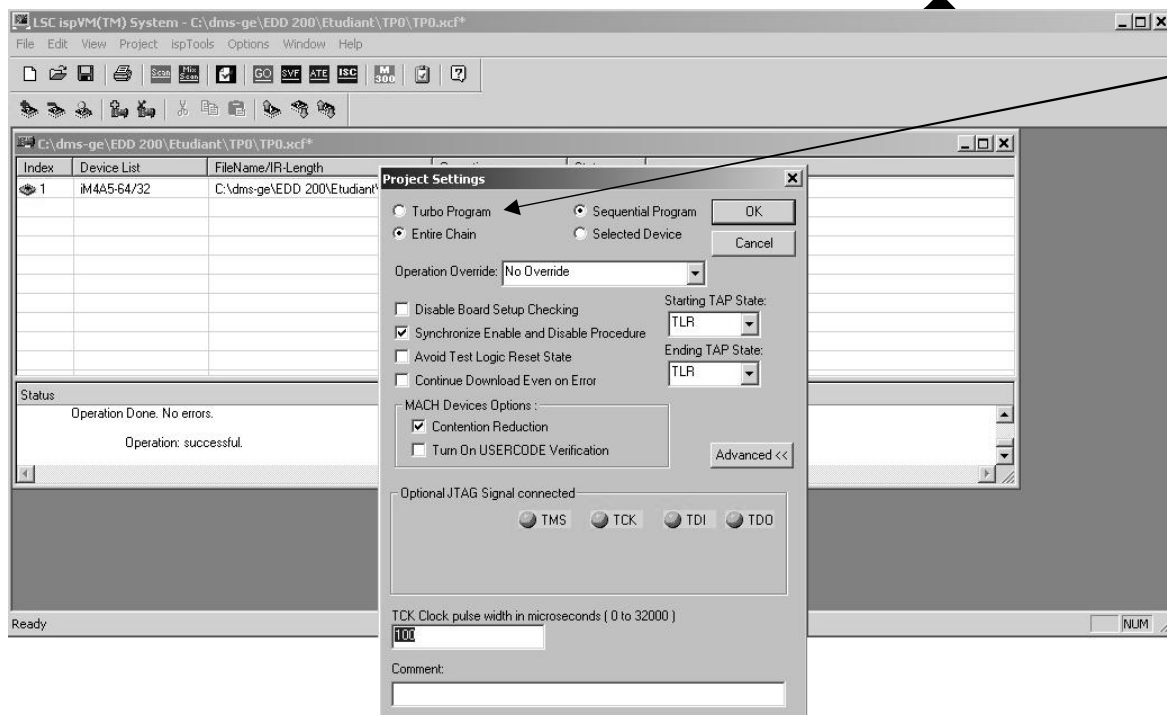
0.12.6 Paramétrage de la vitesse de transmission.

Cliquer sur « **Project** »,

Cliquer sur « **Project Settings...** », puis « **Advanced** »

Implémenter un diviseur d'horloge de 2.

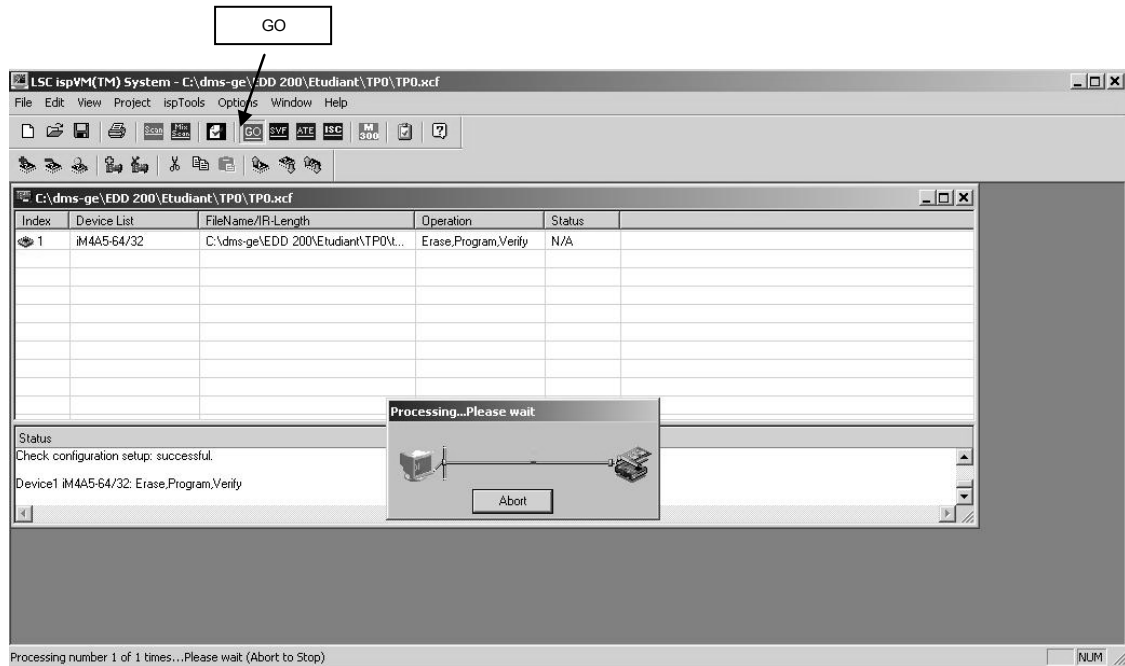
Cliquer sur « **OK** »



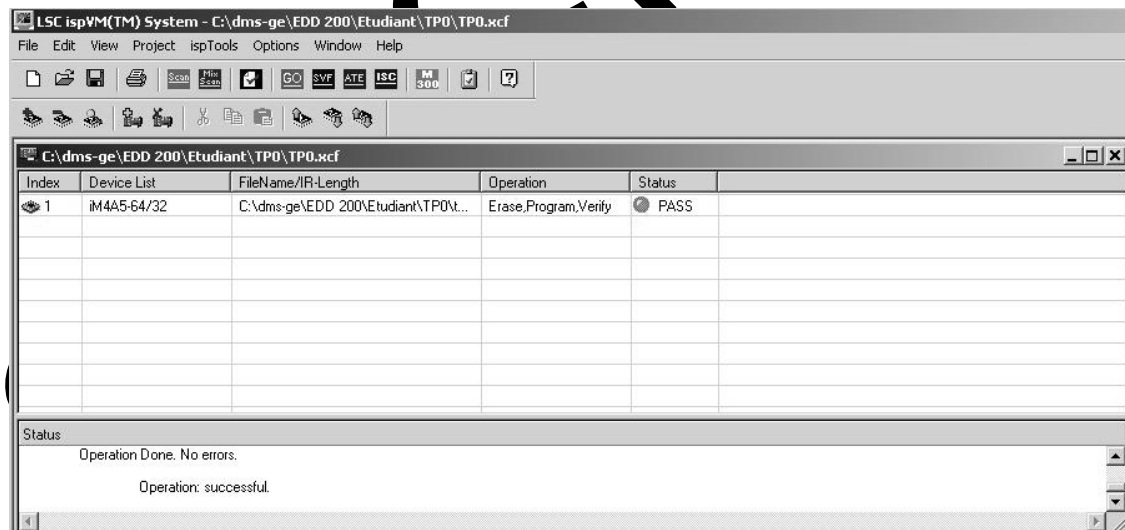
Nota : la vitesse de transmission est le résultat d'une division d'horloge, la qualité de la transmission dépend donc des performances de l'ordinateur. En cas d'anomalies constatées ou de problèmes lors du téléchargement, cliquer l'option Turbo Program et régler le TCK sur 1.

0.12.7 Téléchargement du fichier JEDEC.

Cliquer ensuite sur « **GO** »,

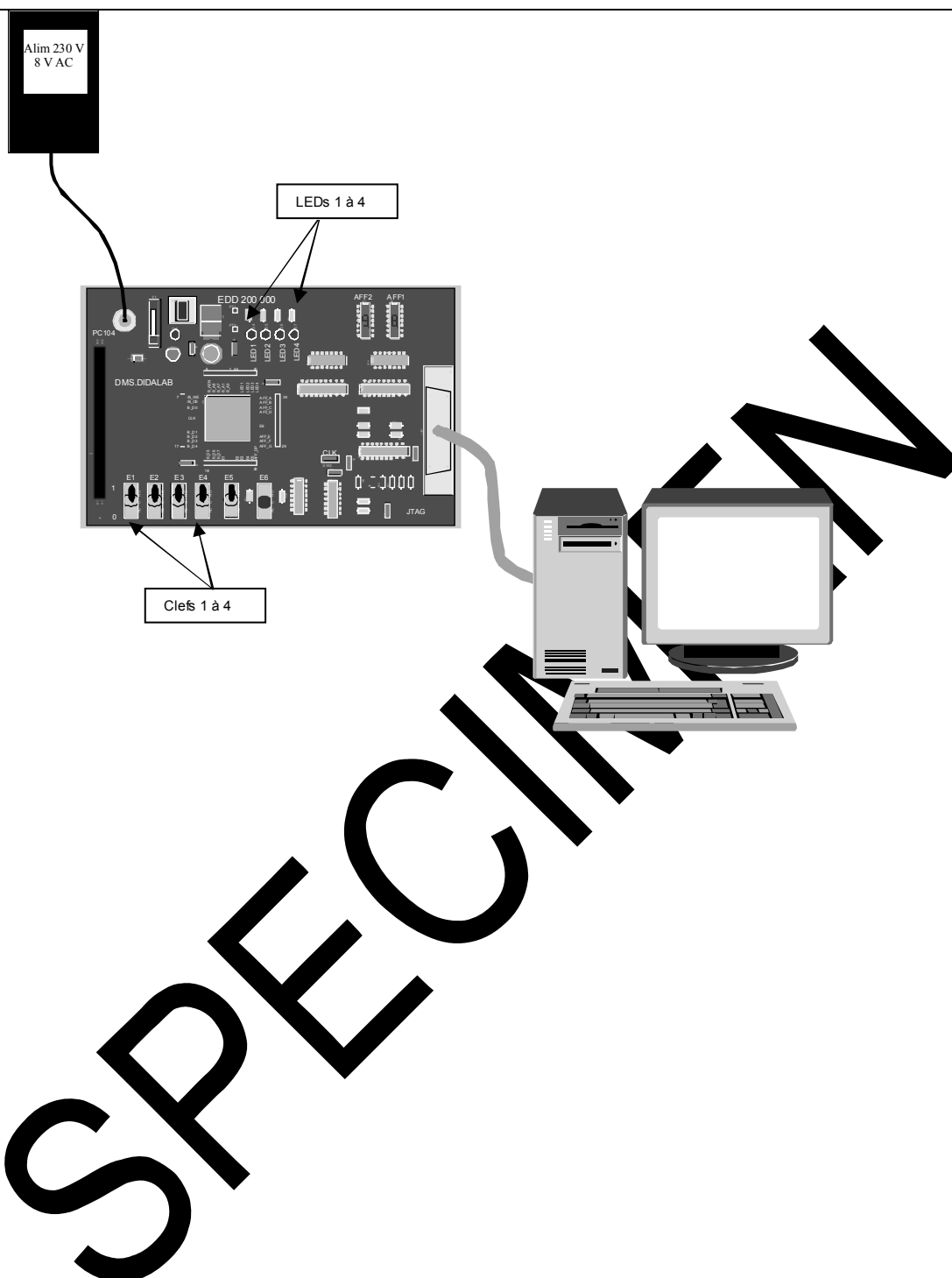


Compte rendu de téléchargement



Vérifier qu'aucune erreur de téléchargement n'est signalée, si oui recommencer l'opération.

Vérifier le fonctionnement du EDD200000, en manoeuvrant les clefs 1 à 4 dont la position doit se recopier sur les LEDs de visualisation 1 à 4.



TP 1 LOGIQUE COMBINATOIRE SIMPLE

1.1 FONCTIONS : ET, NON-ET, OU, NON-OU

Ces fonctions simples sont décrites par un fichier VHDL simple.

Chaque opérateur a deux entrées (les mêmes pour les 4) : E1 et E2.

Les 4 sorties sont visualisées par les 4 LEDs : LED1 à LED4.

Elles sont complémentées pour faire correspondre l'allumage d'une LED à un niveau 1.

SPECIMEN

TP 2 MULTIPLEXEUR

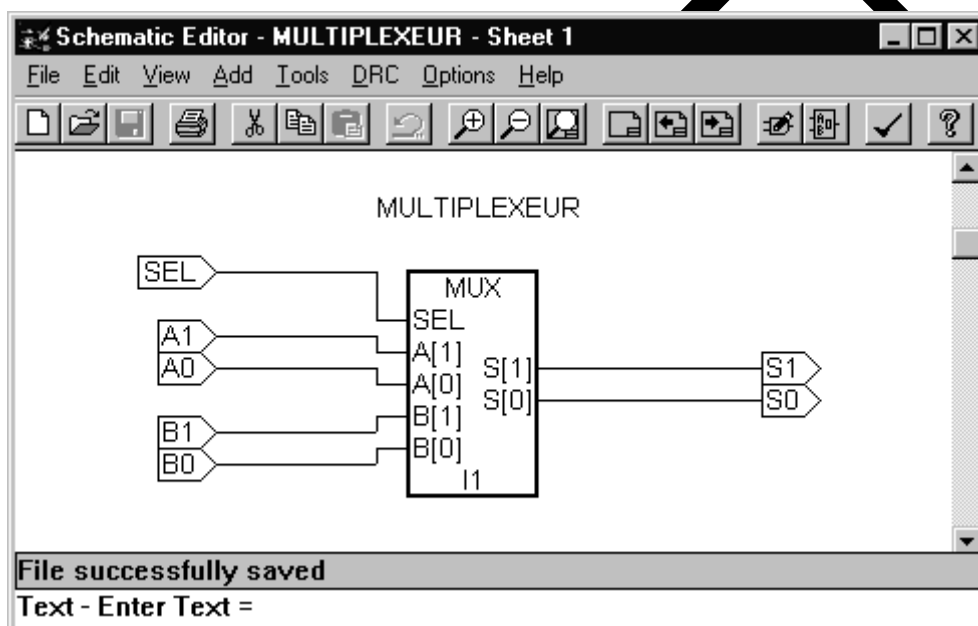
2.1 FONCTION MULTIPLEXEUR

Le projet est de type hiérarchique.

Un module **MULTIPLEXEUR** est réalisé grâce à l'éditeur de schéma ; c'est un simple bloc à 2 entrées de 2 bits : A et B et une 3^{ème} entrée de sélection : SEL.

2.2 LE MODULE HIERARCHIQUE MUX

Il est donné par le schéma de la figure ci-dessous.

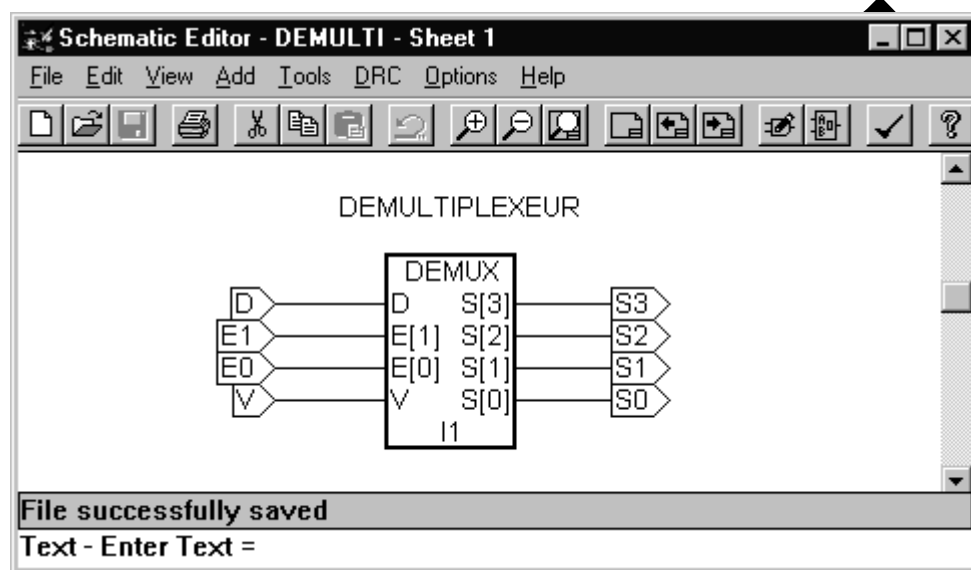


DEMULTIPLEXEUR**2.3 LE DEMULTIPLEXEUR**

Cette fonction réalise l'opération inverse de celle du multiplexeur.

Le projet est de type hiérarchique.

Le module DEMUX est donné ci-dessous.



DECODEUR 7 SEGMENTS

2.4 DECODEUR 7 SEGMENTS

Le projet est de type VHDL seulement

L'entité **SEG** est décrit par un fichier VHDL.

Le décodage s'effectue en 2 parties :

- affichage des nombres de 0 à 9 sur l'afficheur Unités
- affichage des nombres de 10 à 15 (A à F en hexadécimal) sur l'afficheur Dizaines.

2.5 LE FICHIER VHDL

SPECIMEN

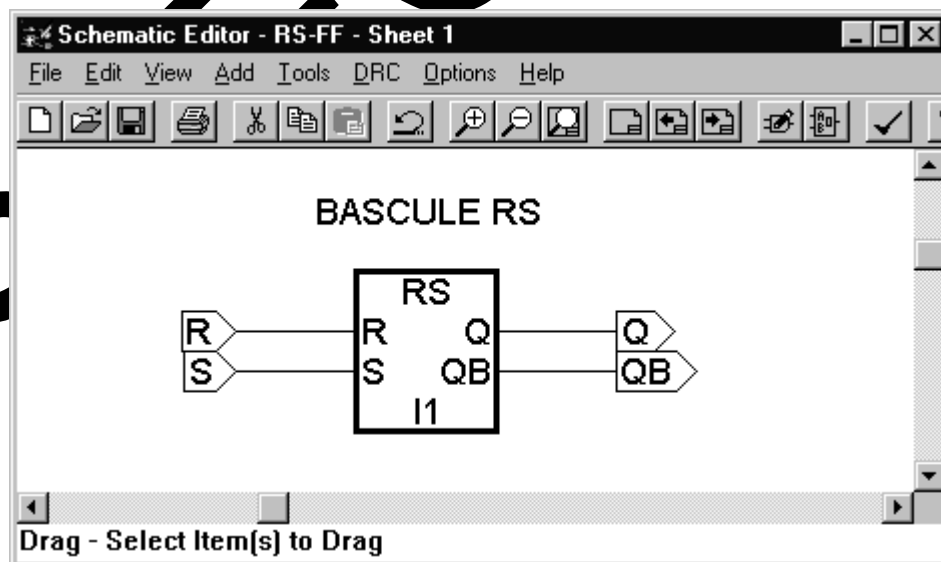
TP 3 BASCULE RS ELEMENTAIRE

3.1 LA BASCULE RS ELEMENTAIRE

La bascule RS est définie par la table de séquence suivante :

Séquence	Entrées		Sorties		Remarques
	R	S	Q_n	Q_{n+1}	
1	0	0	0	0	Maintien à 0
2	0	0	1	1	Maintien à 1
3	0	1	0	1	Mise à 1
4	0	1	1	1	Confirmation du 1
5	1	0	0	0	Confirmation du 0
6	1	0	1	1	Mise à 0
7	1	1	0	Φ	Combinaison de R et S
8	1	1	1	Φ	interdite

Le projet est de type hiérarchique (voir schéma ci-dessous).



BASCULE JK

3.2 LA BASCULE JK

La table de vérité de la bascule JK est la suivante :

↓

↓

↓

↓

H	JK	Q_{n+1}	Commentaires
	00	Q_n	Etat mémoire
	01	0	Mise à 0
	10	1	Mise à 1
	11	$\overline{Q_n}$	Complément de la sortie

Le projet est de type VHDL seulement

TP 4 COMPT. / DECOMPT. BCD et PREDIV. PROGRAMMABLE

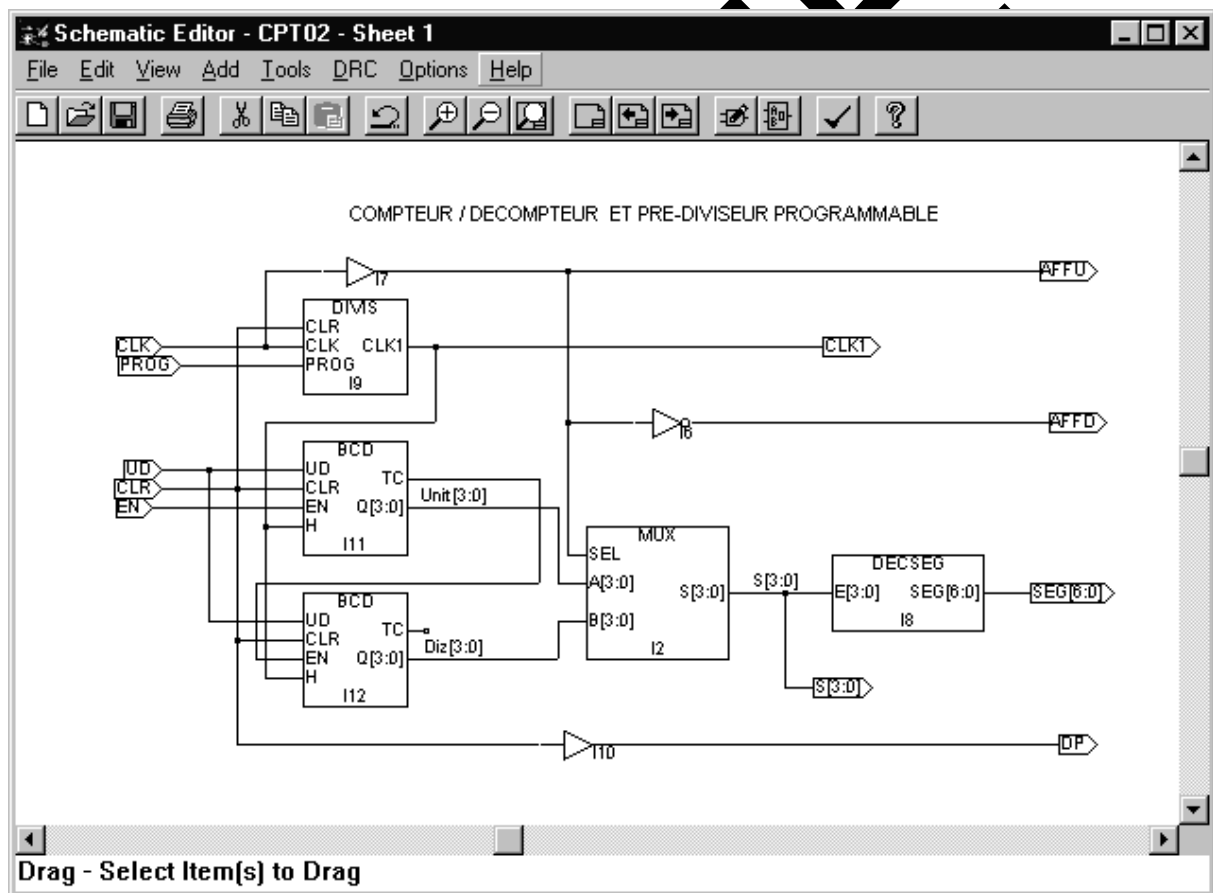
4.1 COMPTEUR / DECOMPTEUR – PREDIVISEUR PROGRAMMABLE

Le projet est de type hiérarchique.

L'avantage de cette méthode réside dans la possibilité d'étudier séparément chaque module fonctionnel du projet, de le tester et surtout de simplifier l'écriture du programme dans la partie interconnexion entre blocs.

4.2 LES MODULES

Le schéma est donné ci-dessous.



4.2.1 PREDIVISEUR