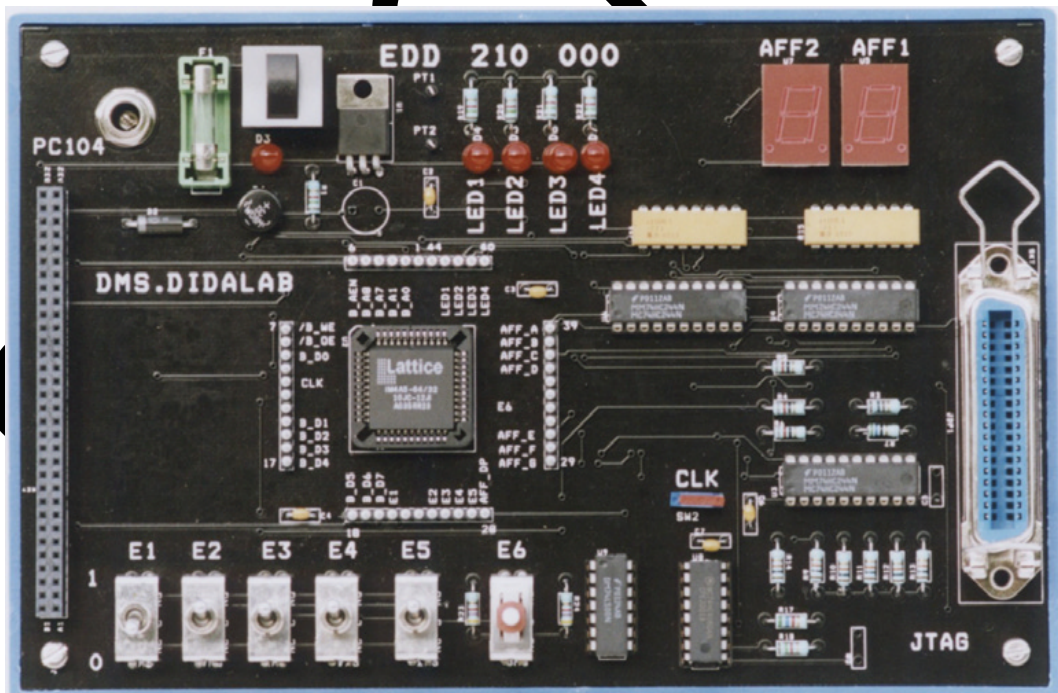


MANUEL PROFESSEUR
TRAVAUX PRATIQUES
LOGIQUE PROGRAMMEE SUR
EPLD

IMMEDIAT



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	37
TP 2	MULTIPLEXEUR	41
TP 3	DEMULTIPLEXEUR	45
TP 4	DECODEUR 7 SEGMENTS	49
TP 5	BASCULE RS ELEMENTAIRE	51
TP 6	COMPT. / DECOMPT. BCD et PREDIV. PROGRAMMABLE	57

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.

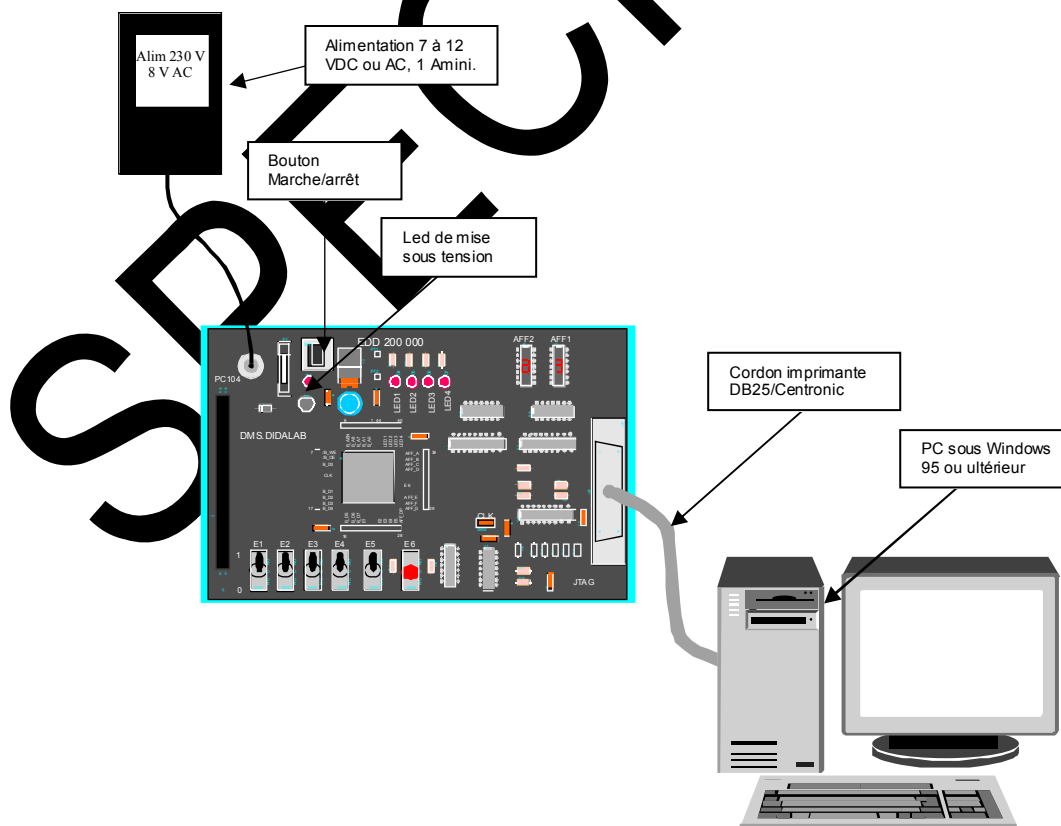
0.1 Avertissement

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. Elle se décompose en deux parties, la première se limite à utiliser une application existante et à la manipuler, la seconde facultative, permet à l'utilisateur de créer son application. Cette seconde partie peut être mise en œuvre soit immédiatement, soit après avoir étudié l'ensemble des travaux pratiques existants.

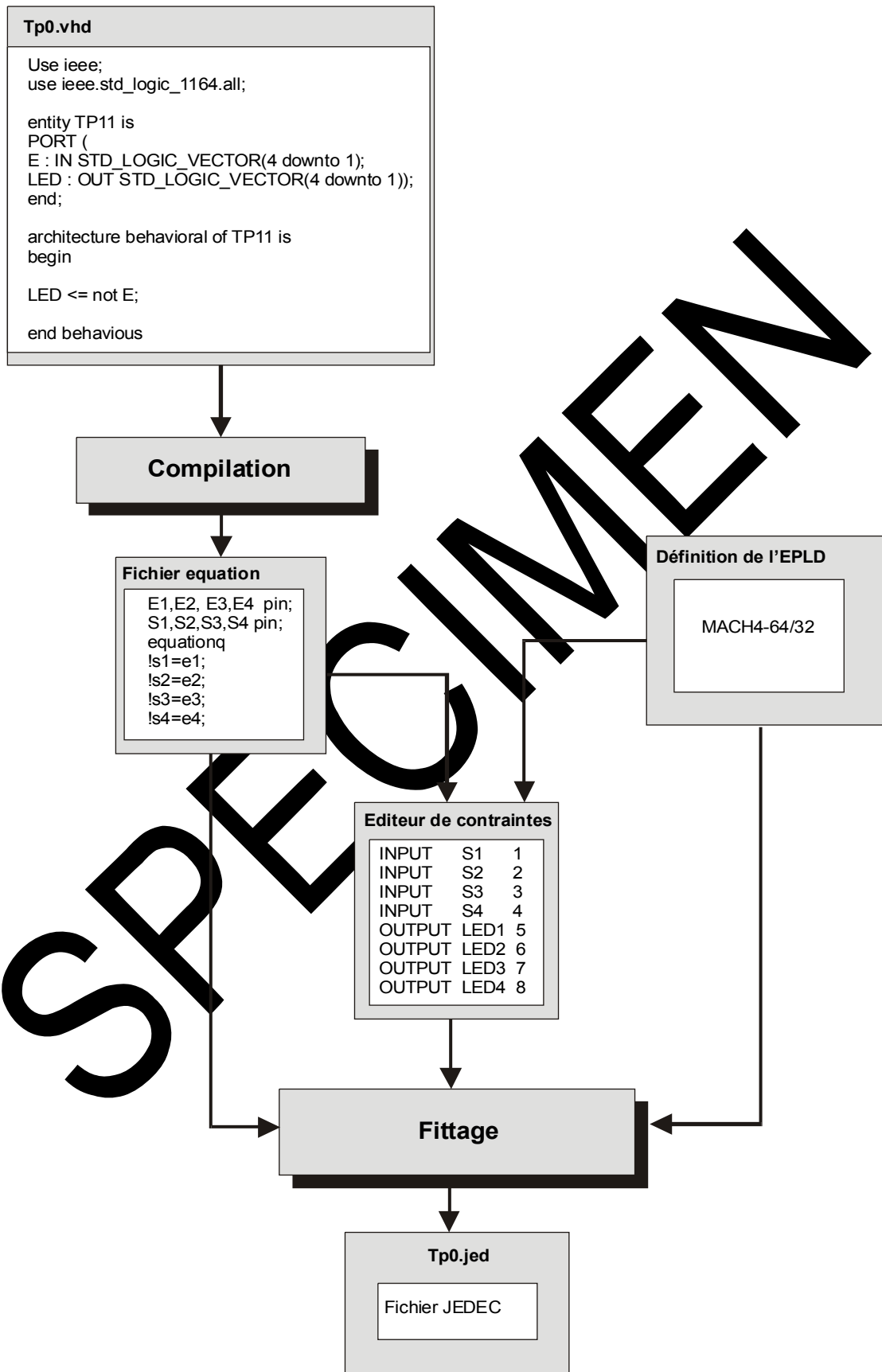
PREMIERE PARTIE

0.2 Installation du matériel.

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



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



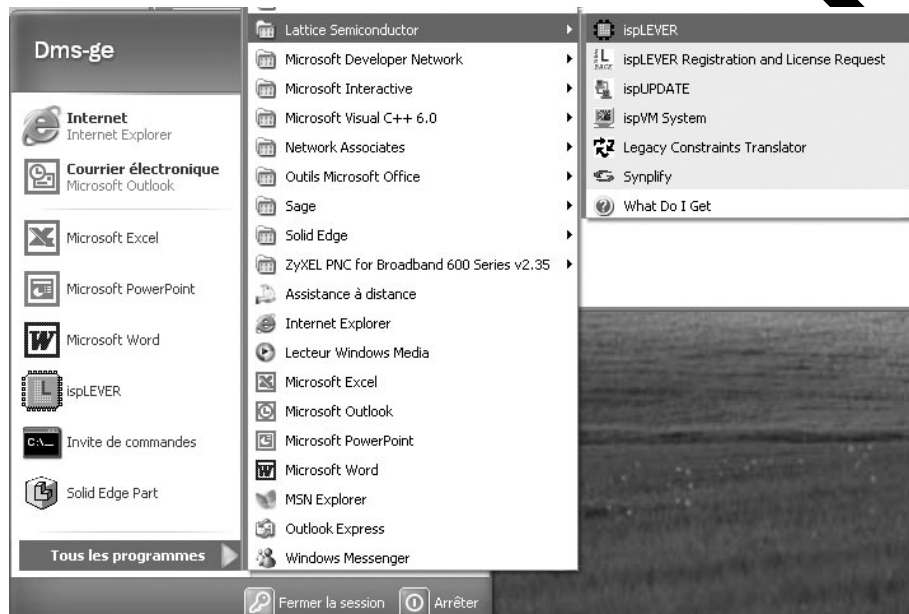
0.4 Démarrage du logiciel.

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** »,



SPEC

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 « **Disque local(C :)** », (par la fenêtre de sélection du chemin),

0.5.4 Cliquer sur « **dms-ge** »,

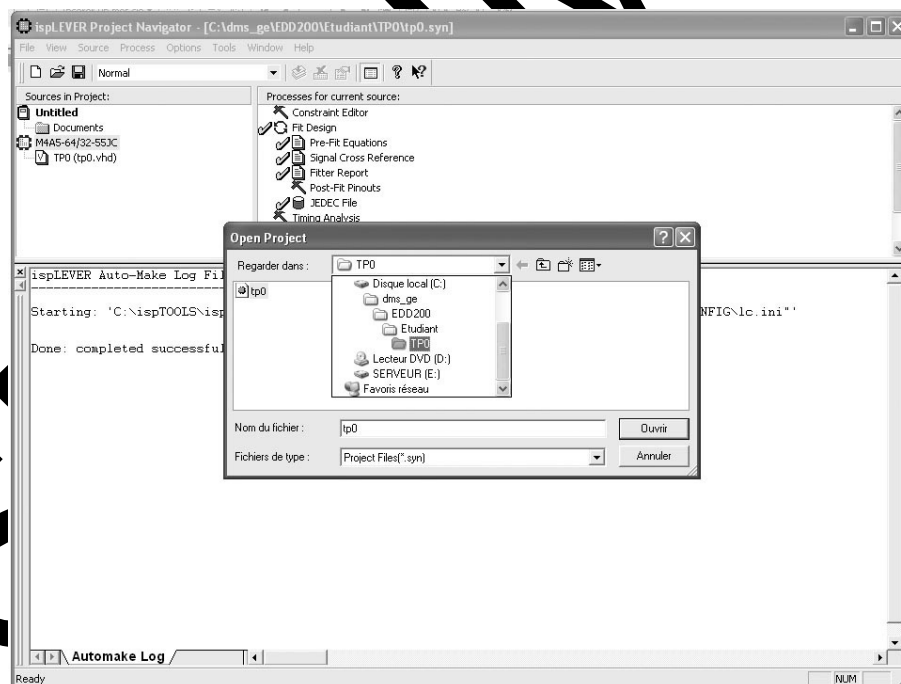
0.5.5 Cliquer sur « **Edd200** »,

0.5.6 Cliquer sur « **Etudiant** »,

0.5.7 Cliquer sur « **tp0** » (répertoire),

0.5.8 Cliquer sur « **tp0** » (fichier),

0.5.9 Cliquer sur « **Ouvrir** »,



0.6 Edition du fichier exemple : Tp0,

0.6.1 Dans la fenêtre « Sources in Project : » :

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



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

entity TP0 is
    PORT (
        E : IN STD_LOGIC_VECTOR(4 downto 1);
        LED : OUT STD_LOGIC_VECTOR(4 downto 1));
end;

architecture behavioral of TP0 is
begin
    LED <= not E;
end behavioral;
```

--- entity *) Déclaration des entrées sorties
--- E1 à E4 en entrées (broches 24 à 27)
--- LEDs 1 à 4 en sorties visualisation
--- Architecture *) portion de description du comportement
--- Begin *) Début de la partie opératoire
--- Instruction not *) inversion de la variable E
--- et sortie du résultat sur le port LED1/4
--- Fin du programme

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

- **Library ieee** : inclusion de la librairie standard
- **Use ieee.std_logic...** : utilisation des ports d'entrées et sorties standards,
- **Entity** : définitions des ports d'entrées et sorties globaux,
- **Architecture** : portion de description du comportement.

SPECIMIN

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 indiquées ci-dessous :

Paramètres à valider :

Family = ispMACH4A5**,

Device = M4A5-64/32,

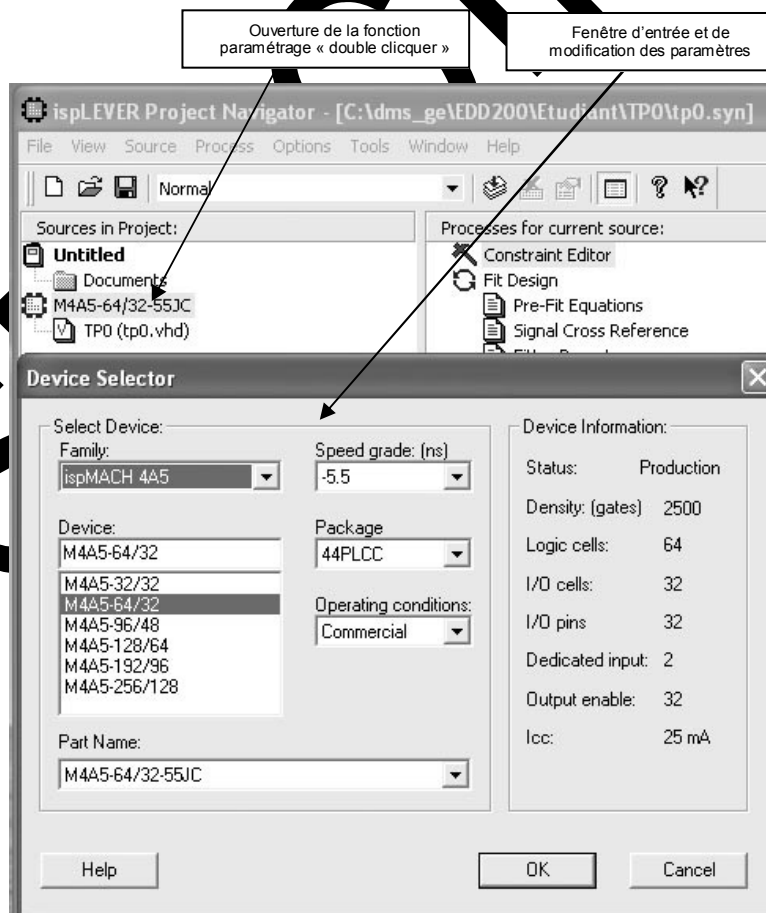
Speed grade (ns) = -5.5

Package = 44PLCC,

Operating condition = commercial,

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

Cliquer sur “OK”

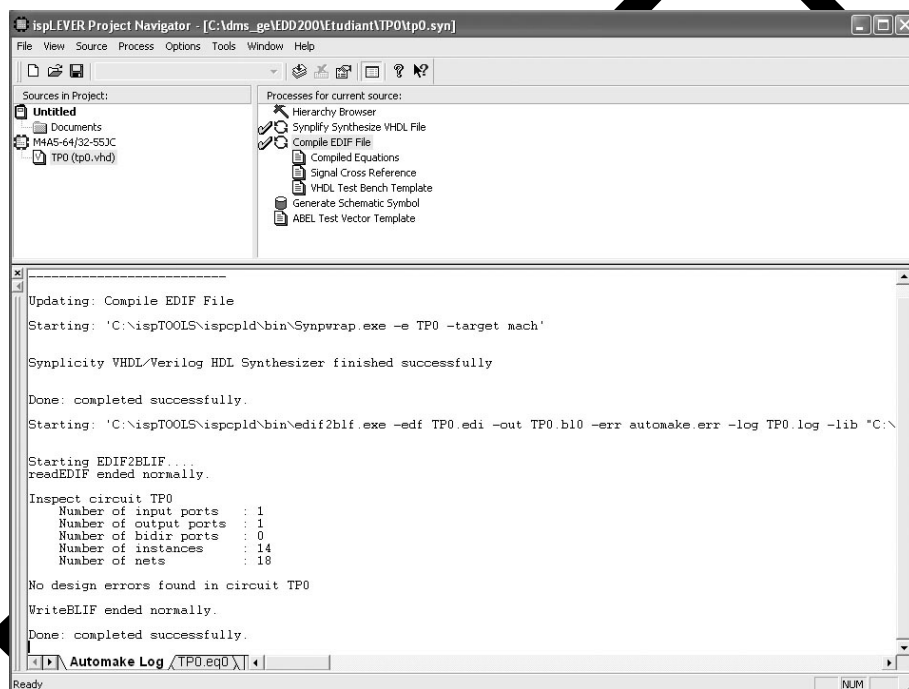


0.8 Compilation du fichier VHDL.

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 sortantes.



```

isplEVER Project Navigator - [C:\dms_geVEDD2001\etudiant\TP0\tp0.syn]
File View Source Process Options Tools Window Help

Sources in Project:
  Untitled
  Documents
  M4AS-64/32-5S3C
  TP0 (tp0.vhd)

Processes for current source:
  Hierarchy Browser
  Synplify Synthesize VHDL File
  Compile EDIF File
  Compiled Equations
  Signal Cross Reference
  VHDL Test Bench Template
  Generate Schematic Symbol
  ABEL Test Vector Template

Updating: Compile EDIF File
Starting: 'C:\nispTOOLS\nispcpld\bin\Synpwrap.exe -e TP0 -target mach'

Synplicity VHDL/Verilog HDL Synthesizer finished successfully

Done: completed successfully.
Starting: 'C:\nispTOOLS\nispcpld\bin\edif2blif.exe -edf TP0.edi -out TP0.bl0 -err automake.err -log TP0.log -lib "C:\nispTOOLS\nispcpld\bin\lib'

Starting EDIF2BLIF....
readEDIF ended normally.

Inspect circuit TP0
Number of input ports : 1
Number of output ports : 1
Number of bidir ports : 0
Number of instances : 14
Number of nets : 18

No design errors found in circuit TP0

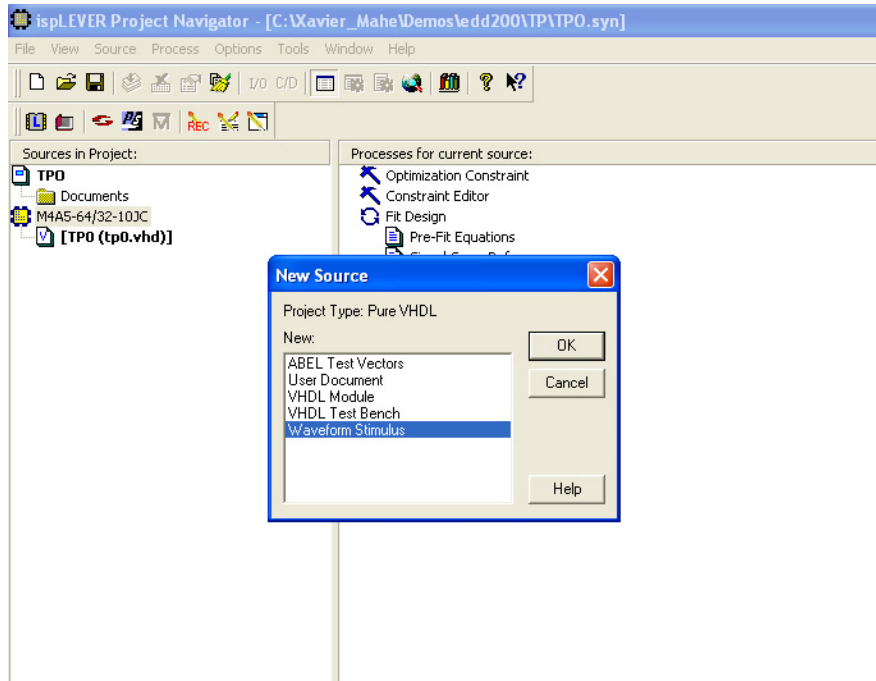
WriteBLIF ended normally.
Done: completed successfully.

Automake Log (TP0.eq0)
Ready
  
```

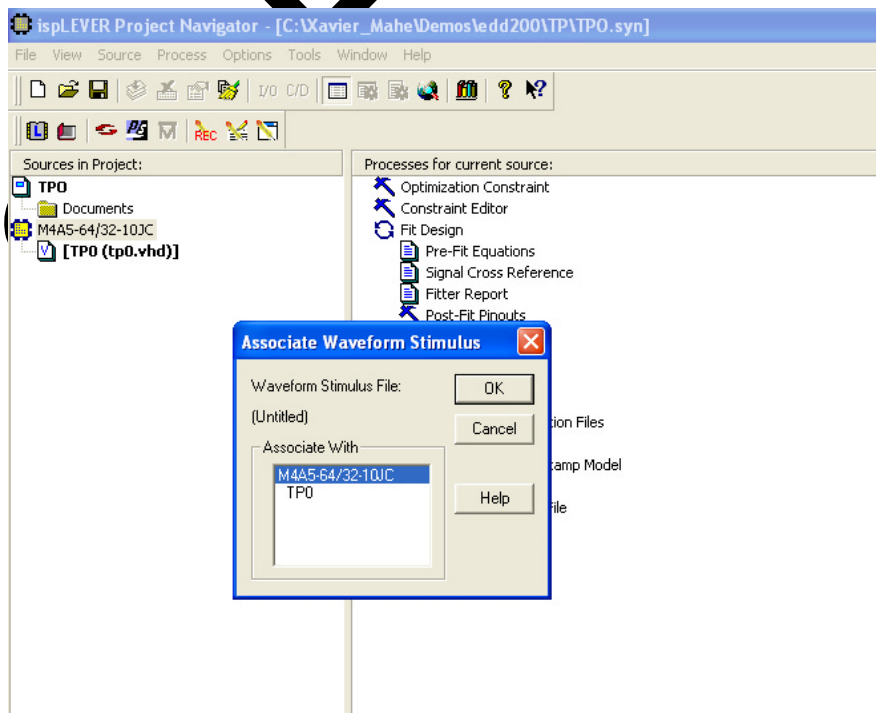
0.9 Simulation de la description VHDL

0.9.1 Création du fichier de simulation

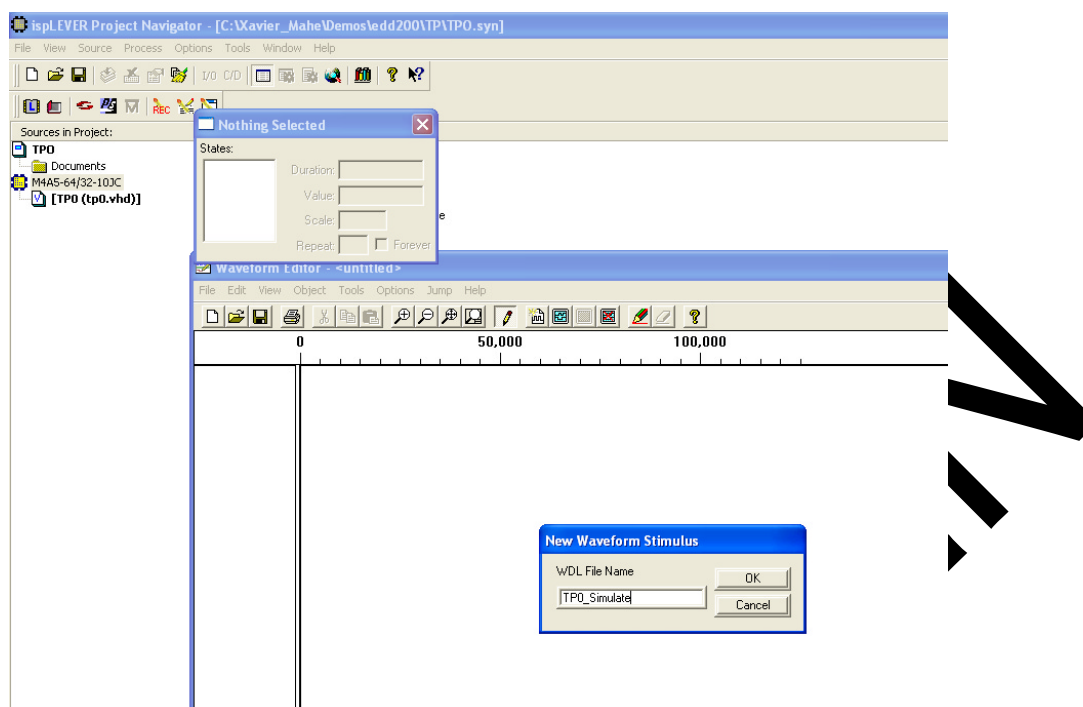
Cliquer droit sur « **M4A5-64/32-10JC** »,
Sélectionner « **new** »
Sélectionner « **waveform stimulus** »



Sélectionner « **M4A5-64/32-10JC** »

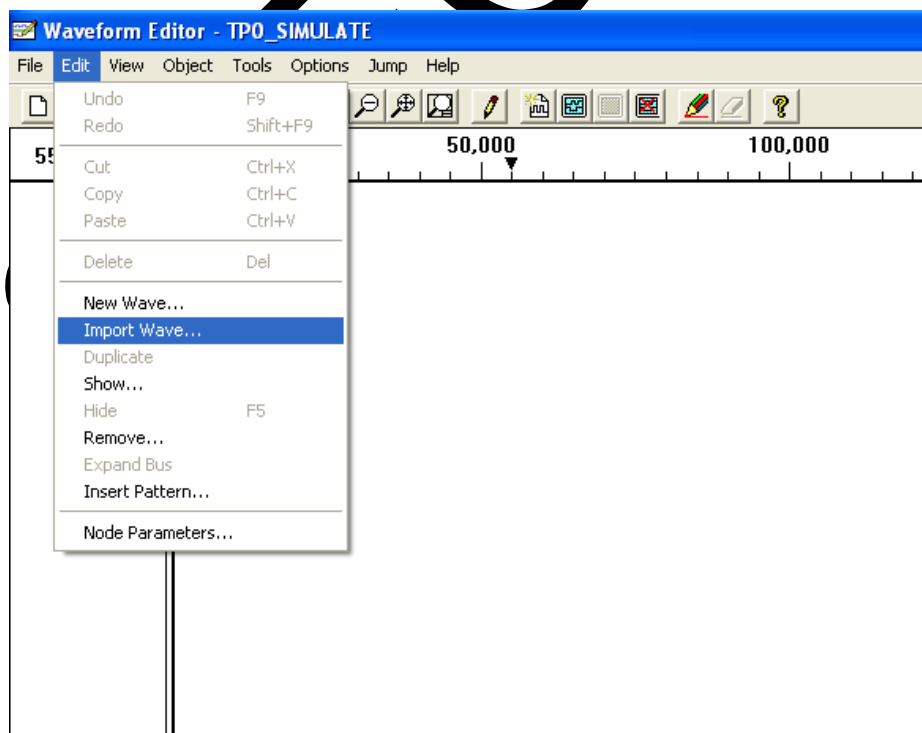


Donner un nom au fichier de simulation (par exemple : « TP0_Simulate »)

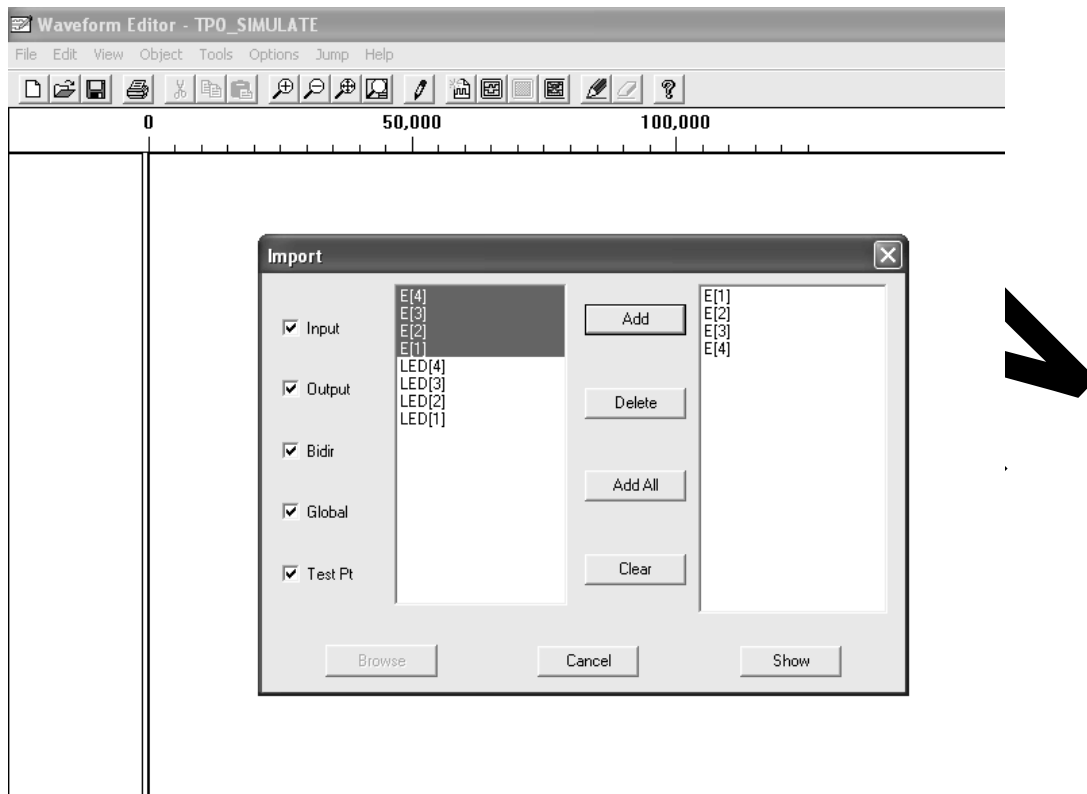


0.9.2 Définition des entrées du système

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



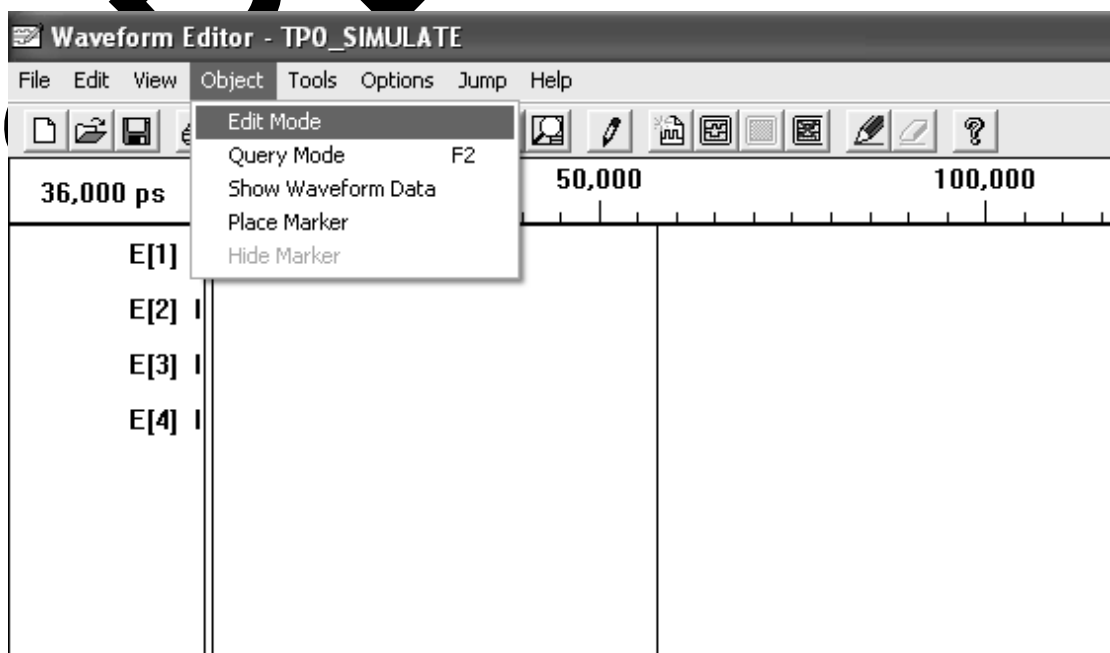
Sélectionner les entrées E(1) à E(4) et cliquer sur « **Add** » afin de les insérer dans la fenêtre d'édition des Stimuli.



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

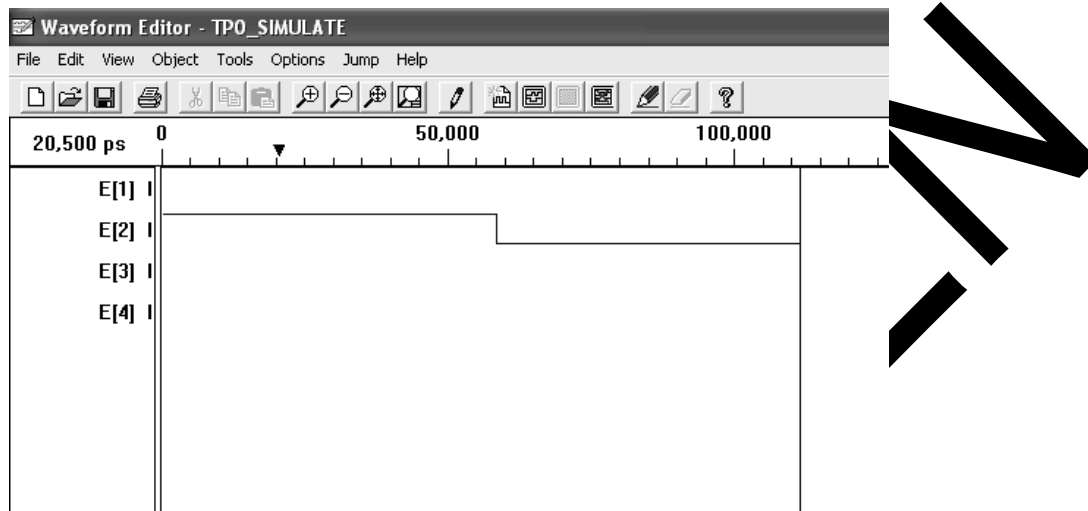
0.9.3 Création des chronogrammes

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



Exemple de création sur E(2) :

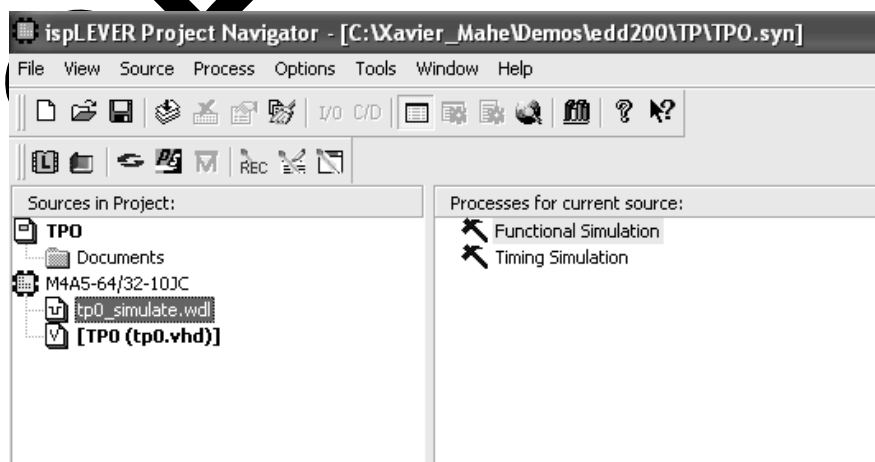
- Cliquer sur la zone graphique à l'horizontale de E(2), faire un glissement horizontal, le chronogramme se positionne par défaut à l'état haut,
- Cliquer ensuite à droite de cette même zone E(2) et faire un glissement, le chronogramme se positionne par défaut à l'état bas,
- etc



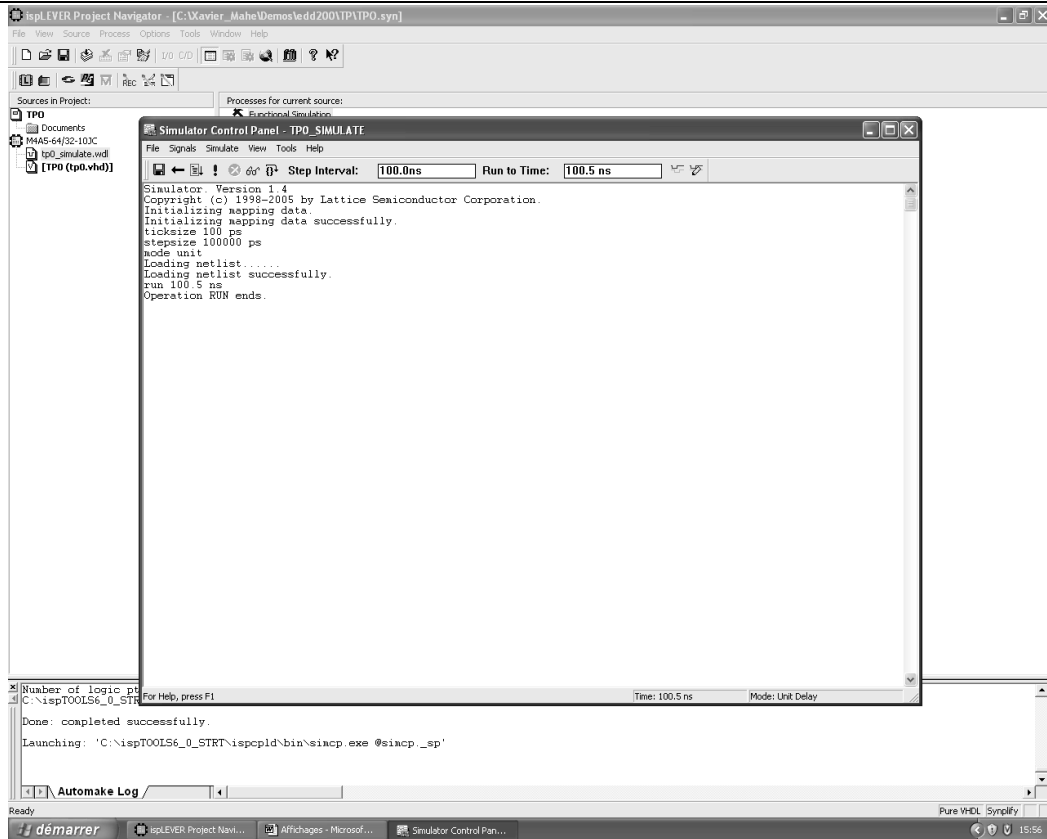
Procéder de même pour chacun des signaux d'entrée
Enregistrer le fichier TPO_Simulate
Sortir du Waveform Editor

0.9.4 Lancement de la simulation

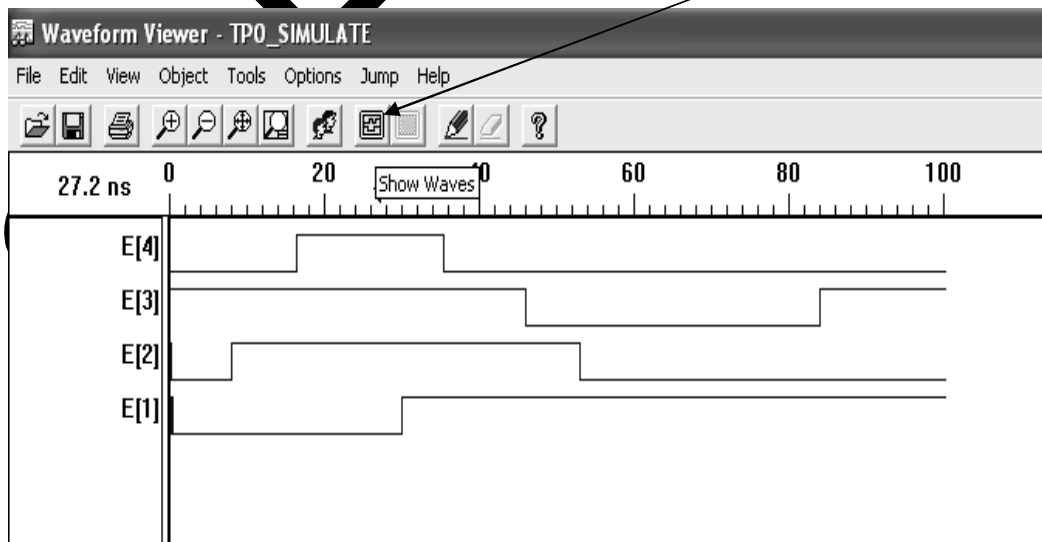
Retourner dans ISLever Project Navigator
Se positionner sur TPO_Simulate.wdl



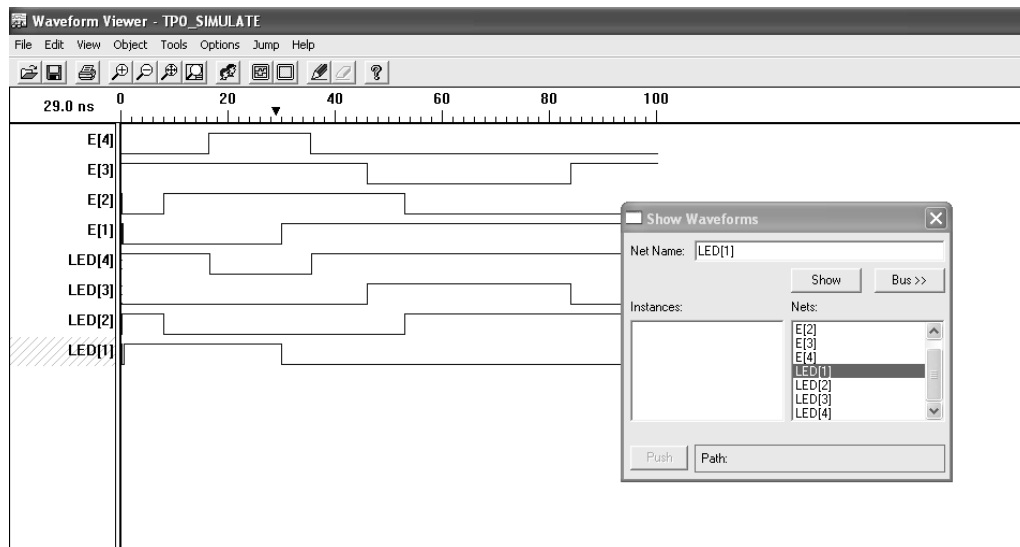
Double-cliquer sur « **Fonctionnal Simulation** »
La fenêtre « Simulator Control Panel » s'ouvre



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



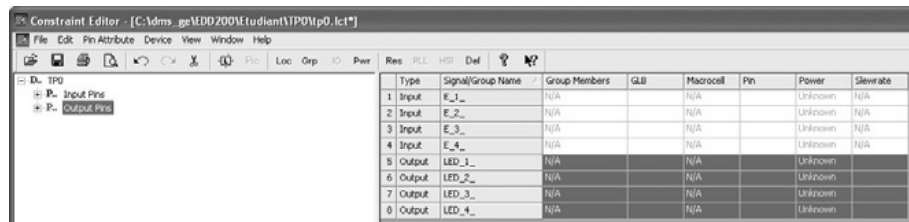
On affiche ainsi le résultat de la simulation

SPECIMEN

0.10 Programmation du brochage du circuit.

0.10.1 Ouverture du fichier

Cliquer sur « **M4A5-64/32-55JC** »,
 Double cliquer sur « **Constraint Editor** »,
 Vérifier les fonctions d'entrée sortie comme indiquées dans la figure ci-dessous.



The screenshot shows the 'Constraint Editor' window for a file named 'C:\dms_ge\EDD2001\Etudiant\TP0\tp0.lci*'. The window contains a table with the following data:

Type	Signal/Group Name	Group Members	GLB	Macrocell	Pin	Power	Slewrate
1 Input	E_1_	N/A		N/A		Unknown	N/A
2 Input	E_2_	N/A		N/A		Unknown	N/A
3 Input	E_3_	N/A		N/A		Unknown	N/A
4 Input	E_4_	N/A		N/A		Unknown	N/A
5 Output	LED_1_	N/A		N/A		Unknown	
6 Output	LED_2_	N/A		N/A		Unknown	
7 Output	LED_3_	N/A		N/A		Unknown	
8 Output	LED_4_	N/A		N/A		Unknown	

SPECIMEN

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

Pour simuler une création de brochage, sous Constraint Editor, Cliquer sur « **Pin Attribut** », puis « **Location Assignment** »

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

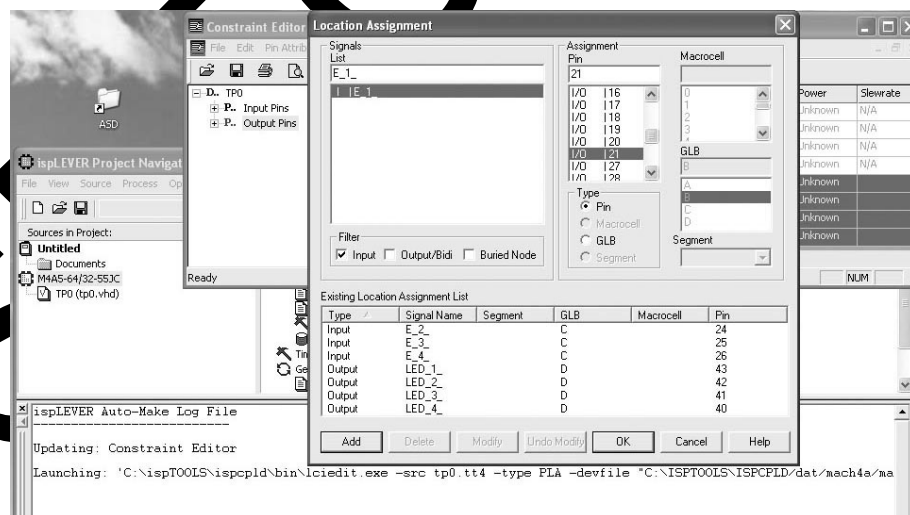
La ligne manquante (broche 21 apparaît dans la fenêtre « **Signals** »).

Pour créer le brochage :

- Déterminer son comportement, dans le cas présent « **Input** »
- Cliquer sur « **I I E_1_** », dans la fenêtre « **Signals List** » pour le valider dans la ligne supérieure comme ci-dessous.
- Cliquer sur « **I/O I21** », dans la fenêtre « **Assignment Pin** »,
- Cliquer sur « **Add** »,
- Cliquer sur « **OK** »

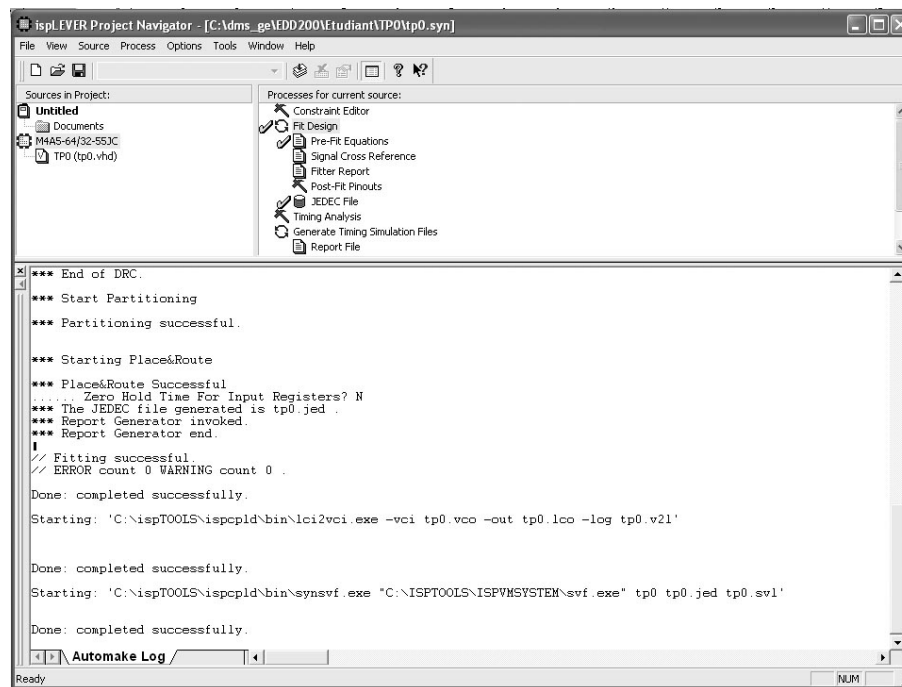
Nota : une sortie serait en réalité une « **output/Bidi** »

Valider et quitter le programme (fermer le logiciel « **Constraint Editor** »)



0.11 FITTAGE

0.11.1 Double Cliquer sur « Fit Design »

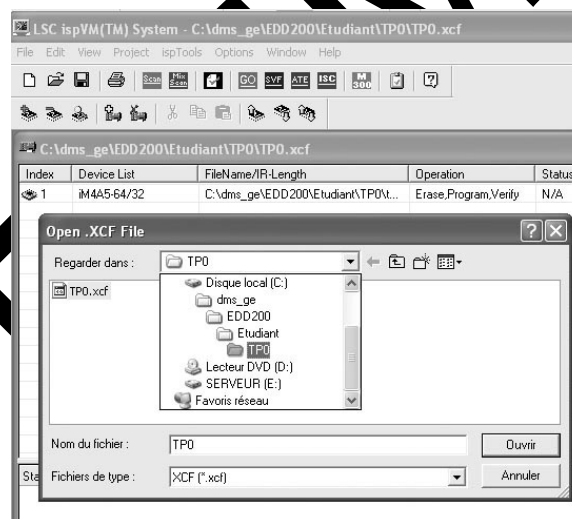


- Sauvegarder en cliquant sur « File », puis « Save », fermer « ips LEVER ».

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** ».0.12.6 Cliquer sur « **File** ».0.12.7 Cliquer sur « **Open** », puis son chemin d'accès.

0.12.8 Paramétrage de la vitesse de transmission.

Après avoir ouvert le projet « **TP0** »

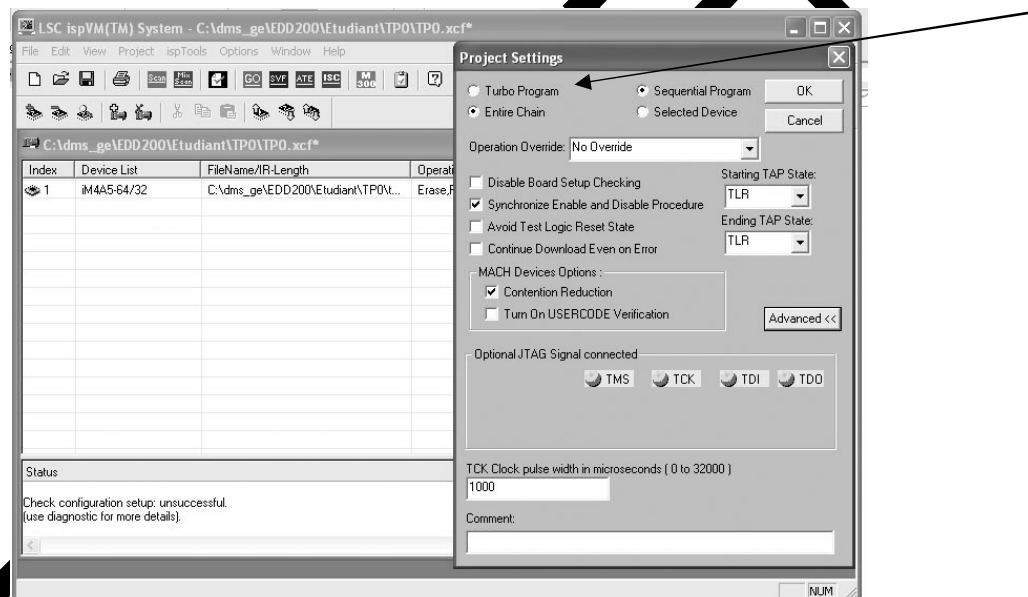
Cliquer sur « **Project** »,

Cliquer sur « **Project Settings**»,

Cliquer sur « **Advanced** »

Implémenter un diviseur d'horloge de 1 à 10 en fonction de la puissance de votre ordinateur, essayer en premier 2 si la transmission s'effectue correctement et conserver cette valeur.

Cliquer sur « **OK** »



Noter que 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 la valeur de TCK sur 1.

0.12.9 Téléchargement du fichier JEDEC.

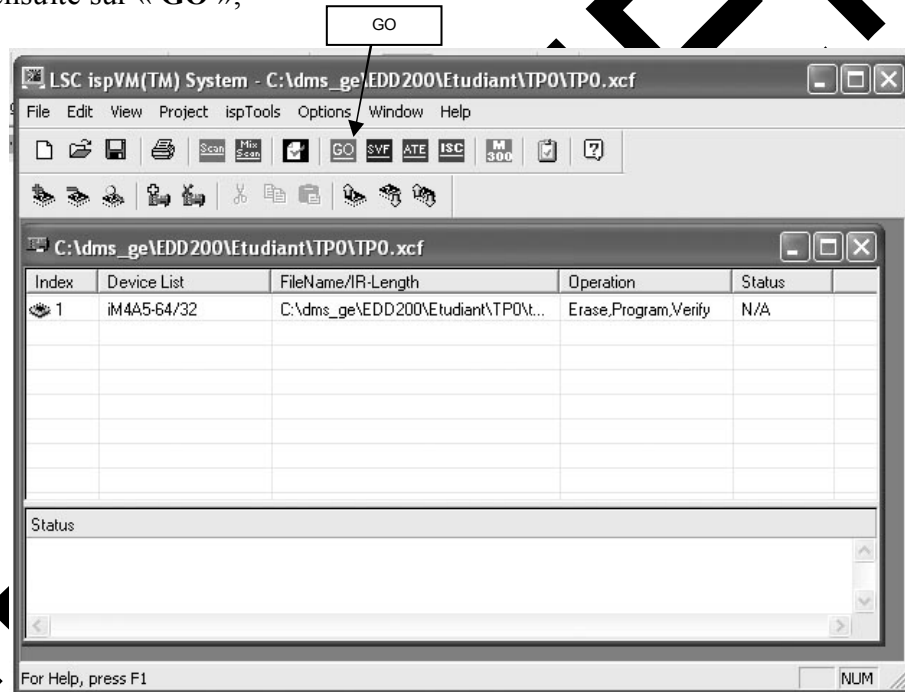
Si le composant a déjà été utilisé, le logiciel pose la question 2 fois,
(XXXXX. Exist. Owerwrite ?), Répondre OK,

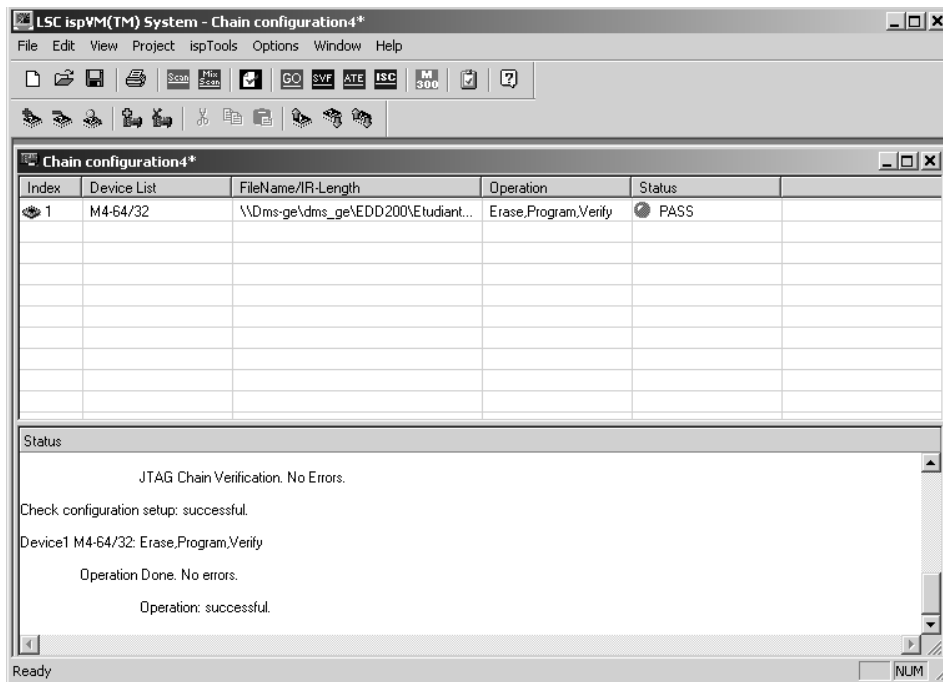
Dans le device list, assurez vous que le composant sélectionné correspond à celui implanté sur la carte (autre exemple de composant possible : M4-64/32) voir p22,

Si le chemin d'accès par défaut à l'installation n'a pas été choisi, vérifier le chemin d'accès dans « **FileName/IR-Lengh** »,

Pour le modifier : Cliquer sur « **Browse** » pour sélectionner le chemin d'accès au fichier jedec à télécharger « **TP0.jed** ».

Cliquer ensuite sur « **GO** »,



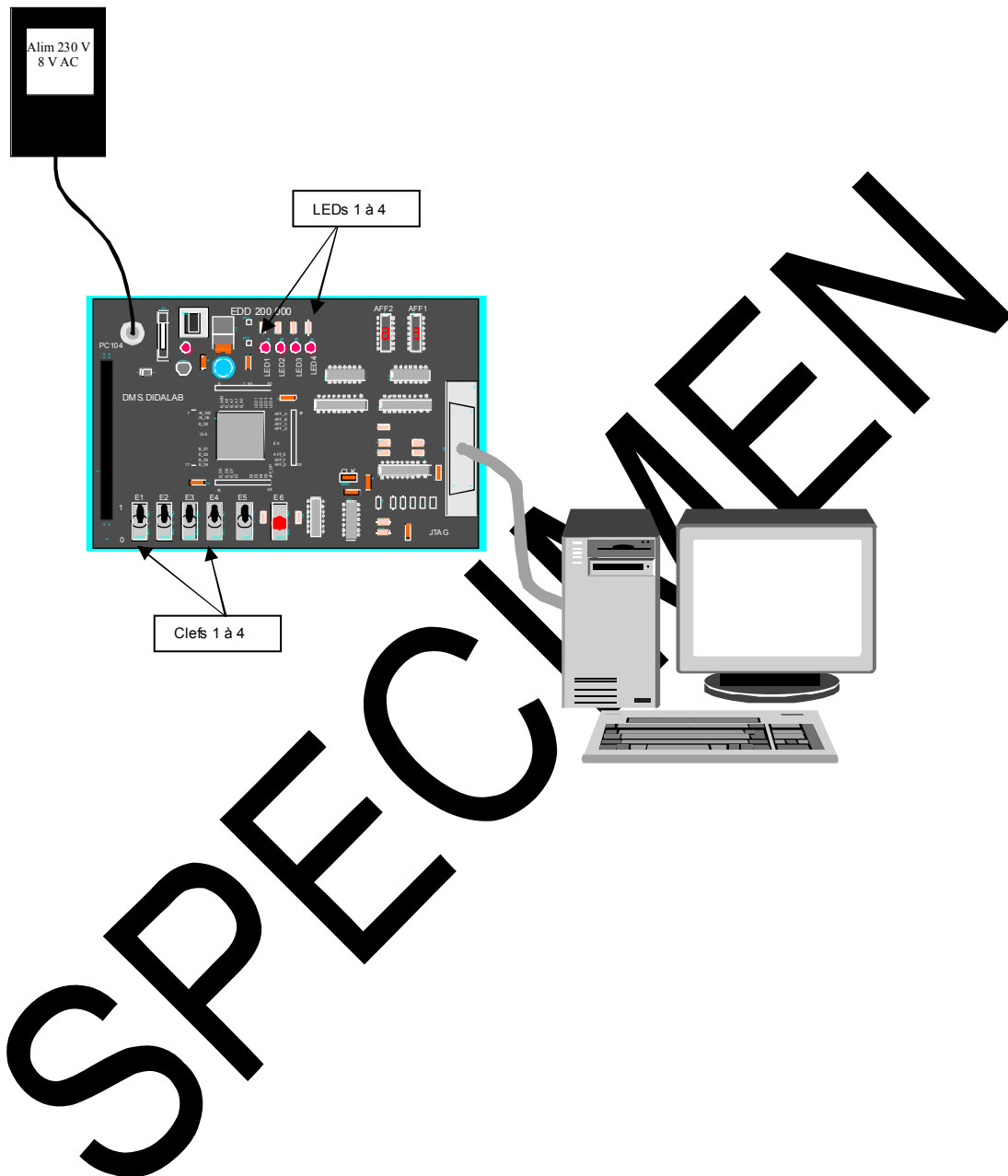


Compte rendu de téléchargement :

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

Si le message « Download Cable not connected to // port address 0x0378 » apparaît se reporter à la configuration du câble page 25

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



DEUXIEME PARTIE (FACULTATIVE)

0.13 Création complète d'un projet :

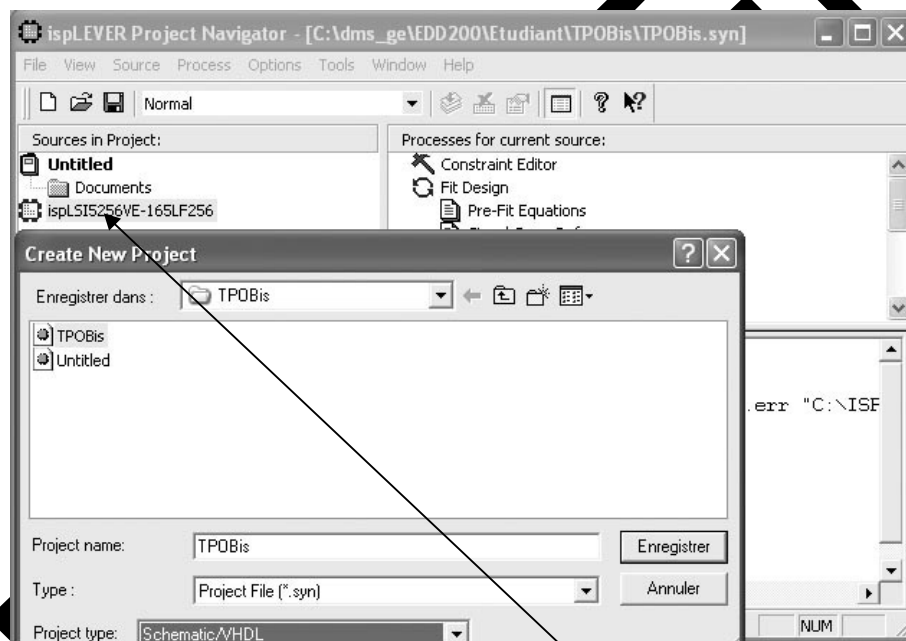
Nota : afin de bien posséder toutes les étapes de création d'un dossier, nous conseillons à nos utilisateurs de réaliser un projet simple.

- Par exemple créer un nouveau dossier qui sera nommé TPOBis, dans un nouveau répertoire TPOBis

0.14 Démarrer le logiciel isp LEVER (voir chapitre 0.4)

0.15 Créer un nouveau projet,

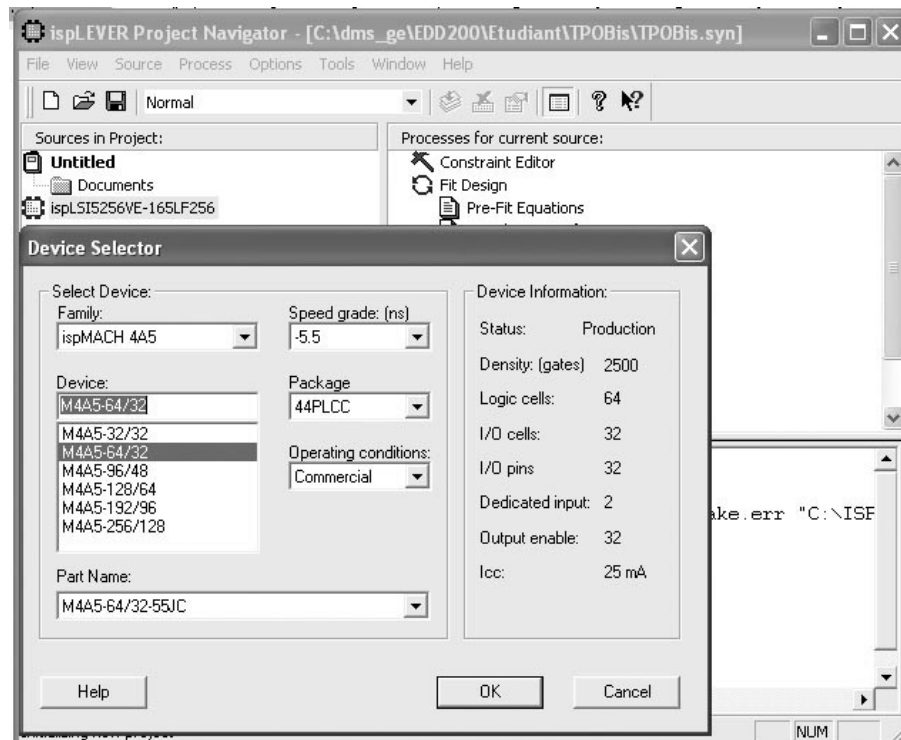
Cliquer sur « File », puis new « Project », entrer le nom du fichier « TPOBis » et le Project type « **schematic/VHDL** ».



Double Clic sur ispSI5256-165LF256,

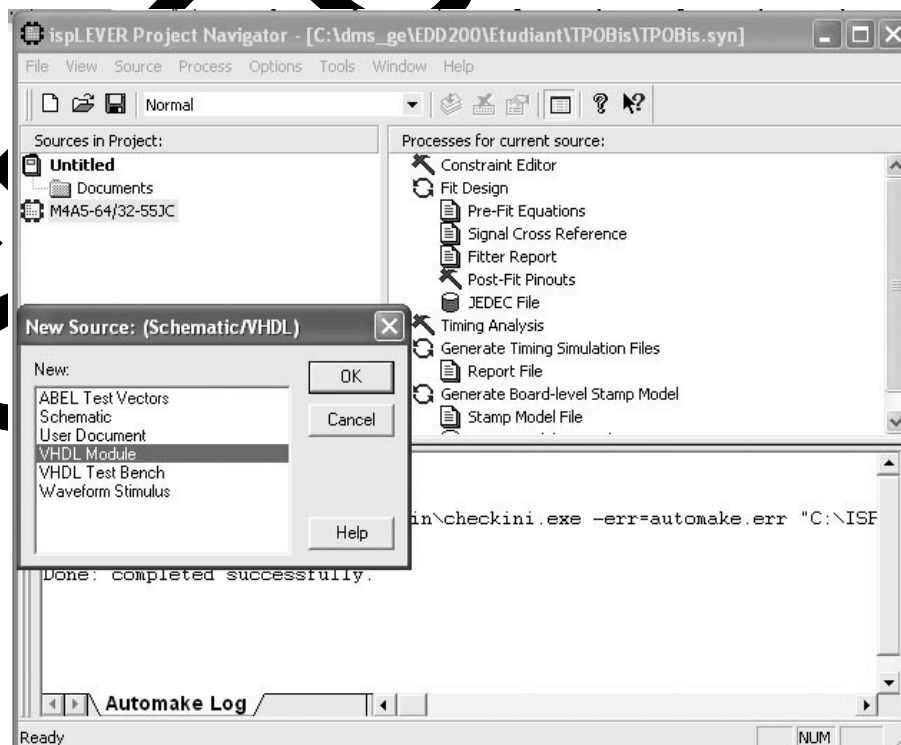
Double Clic

Paramétrer le composant Family « **ispMACH4A5** » puis « Device « **M4A5-64/32** », conformément au composant utilisé sur la carte EDD100000, puis Package « **PLCC44** »



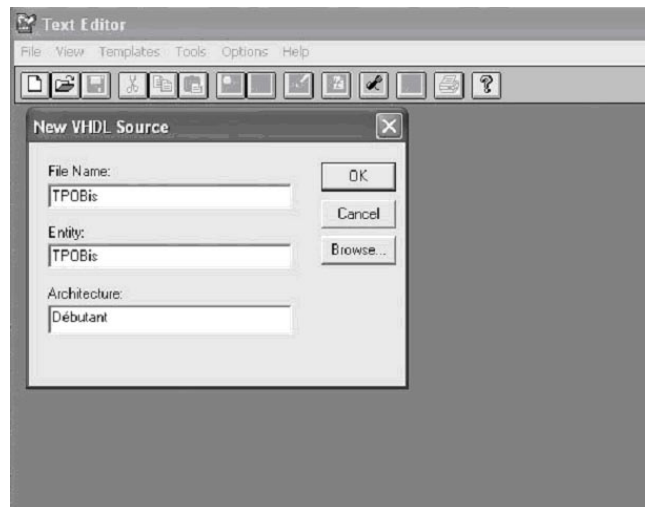
Valider.

Créer le nouveau projet, cliquer sur « **Source** », puis « **New** »,



Sélectionner l'option « **VHDL Module** », valder.

Sélectionner File Name « **TP0Bis** », Entity « **TP0Bis** », Architecture « **Debutant** »,



Cliquer sur « OK »,

Saisir les fonctions rigoureusement comme indiqué ci-dessous **ne surtout pas utiliser les accents**

Afin d'éviter toute erreur de saisie, vous pouvez aussi ouvrir le fichier fourni par nos soins « **TP0.vhd** » et faire un copier coller dans votre nouveau fichier.

```

library ieee;
use ieee.std_logic_1164.all;

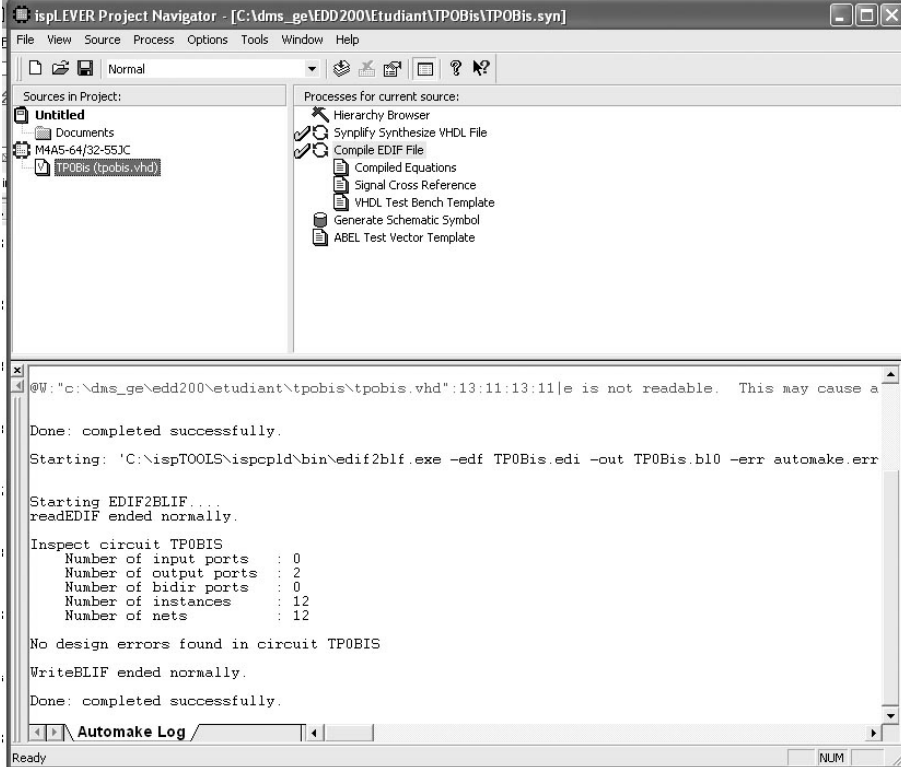
entity TP0Bis is
  PORT (
    E : IN STD_LOGIC_VECTOR(4 downto 1);
    LED : OUT STD_LOGIC_VECTOR(4 downto 1));
end;

architecture Debutant of TP0Bis is
begin
  LED <= not E;
end Debutant;

```

Cliquer sur « **File** », « **Save** », fermer la fenêtre,

Double cliquer sur « **Compile EDIF file** », si vous n'avez commis aucune erreur, vous obtenez l'écran ci-dessous et vous êtes prêt à télécharger le fichier.



```
ispLEVER Project Navigator - [C:\dms_ge\EDD200\Etudiant\TP0Bis\TP0Bis.syn]
File View Source Process Options Tools Window Help

Sources in Project:
- Untitled
- Documents
- MHAS-64/32-55JC
- TP0Bis (tpobis.vhd)

Processes for current source:
- Hierarchy Browser
- Synplify Synthesize VHDL File
- Compile EDIF File
- Compiled Equations
- Signal Cross Reference
- VHDL Test Bench Template
- Generate Schematic Symbol
- ABEL Test Vector Template

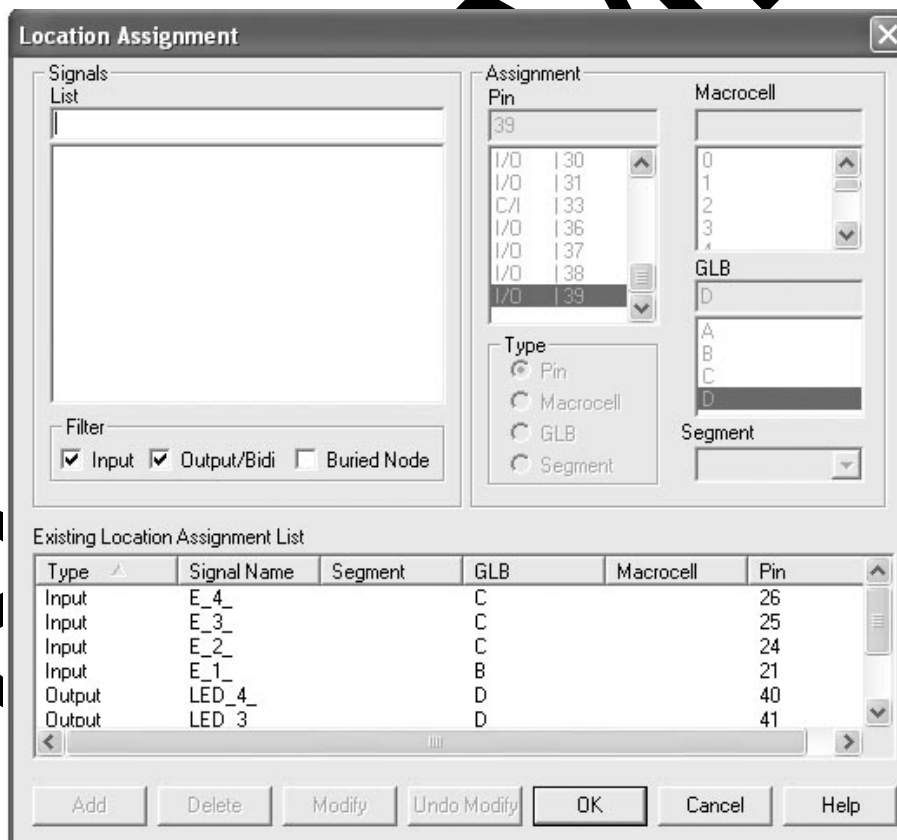
@W: "c:\dms_ge\edd200\etudiant\tpobis\tpobis.vhd":13:11:13:11|e is not readable. This may cause a
Done: completed successfully.
Starting: 'C:\ispTOOLS\isppld\bin\edif2blif.exe -edf TP0Bis.edi -out TP0Bis.bl0 -err automake.err
Starting EDIF2BLIF...
readEDIF ended normally.
Inspect circuit TP0BIS
Number of input ports : 0
Number of output ports : 2
Number of bidir ports : 0
Number of instances : 12
Number of nets : 12
No design errors found in circuit TP0BIS
WriteBLIF ended normally.
Done: completed successfully.
Automake Log /
Ready
```

SPEC

Cliquer sur « **M4A5-64/32-55JC** »,
 Double cliquer sur « **Constraint Editor** »,
 Création des brochages, (nota : il est important de bien respecter les brochages indiqués sur la figure ci –dessous).
 Cliquer« **Pin Attribut** », puis « **Location Assignment** »

- Déterminer son comportement, dans le cas présent « **Input** »,
- Cliquer sur « **I I E_1_** », dans la fenêtre « **Signals List** »pour le valider dans la ligne supérieure comme ci-dessous.
- Cliquer sur « **I/O 21** », dans la fenêtre « **Assignment Pin** »
- Cliquer sur « **Add** » pour valider l'entrée ou la sortie passer au signal suivant,
- Valider les 8 entrées sorties comme ci-dessous.

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



Valider et quitter le programme. (fermer le logiciel « **.IspDesignEXPERT** »)

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

0.16.1 Préparation du téléchargement.

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

0.16.2 Cliquer sur « **démarrer** ».

0.16.3 Cliquer sur « **programme** ».

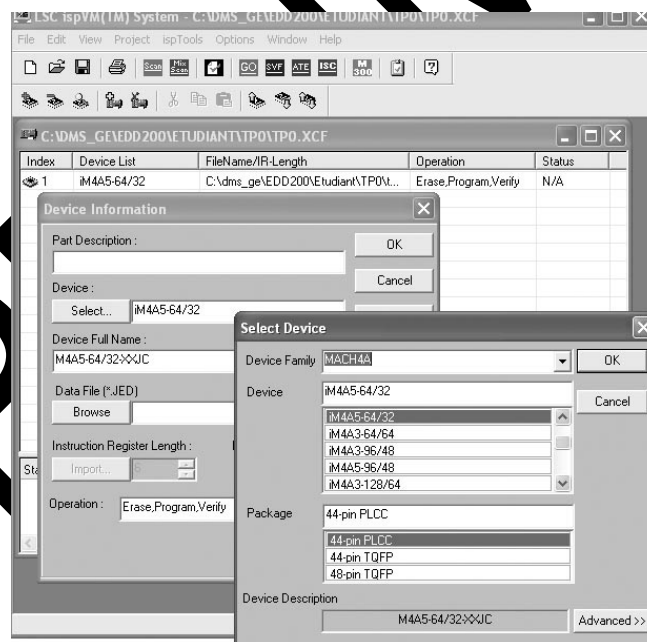
0.16.4 Cliquer sur « **Lattice Semiconductor** ».

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

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

0.16.7 Cliquer sur « **Add Device** ».

0.16.7.1 Cliquer sur « **Select** ».

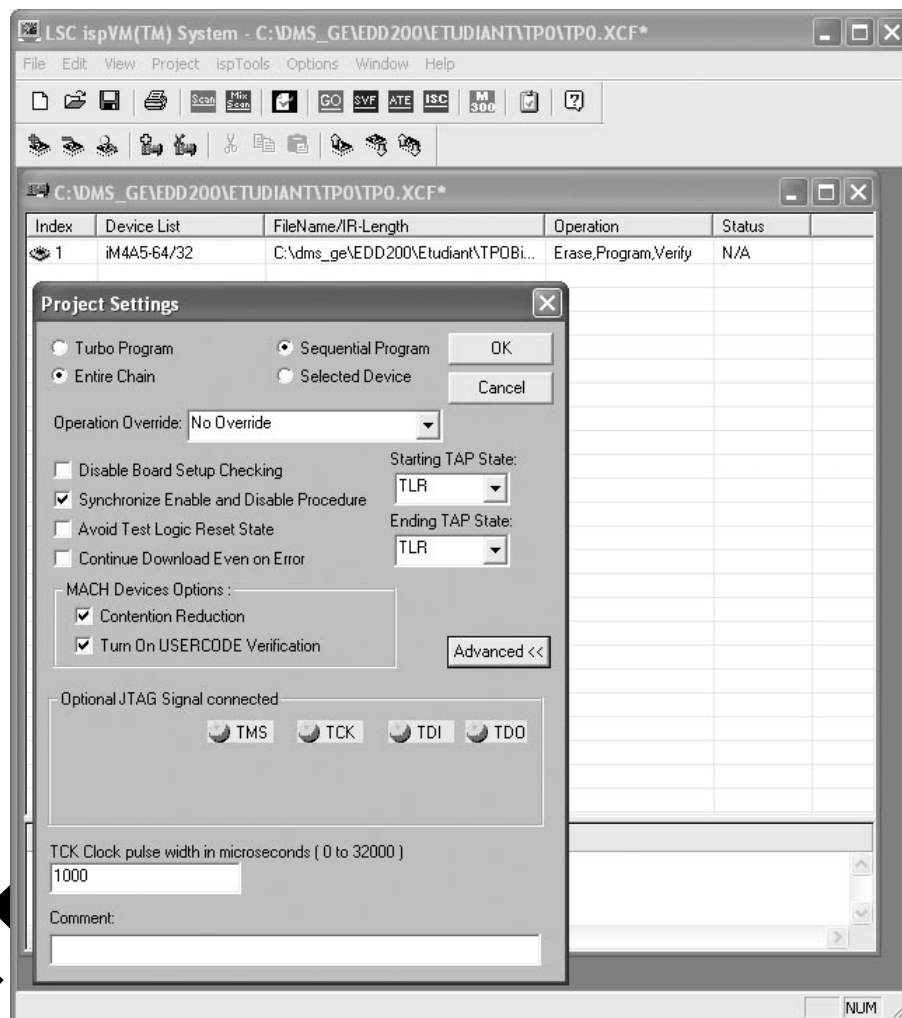


0.16.8 Sélectionner Device Family « **MACH4A** », Device « **iM4A5-64/32** », dans Operation « **Erase,Program,Verify** », Valider,

Nota : vérifier la référence du composant implanté sur la carte, si par exemple il est implanté un M4-64/32, il faut impérativement sélectionner cette famille dans la fenêtre ci-dessus. (choisir Device family « **Mach4** », puis device « **M4-64/32** »).

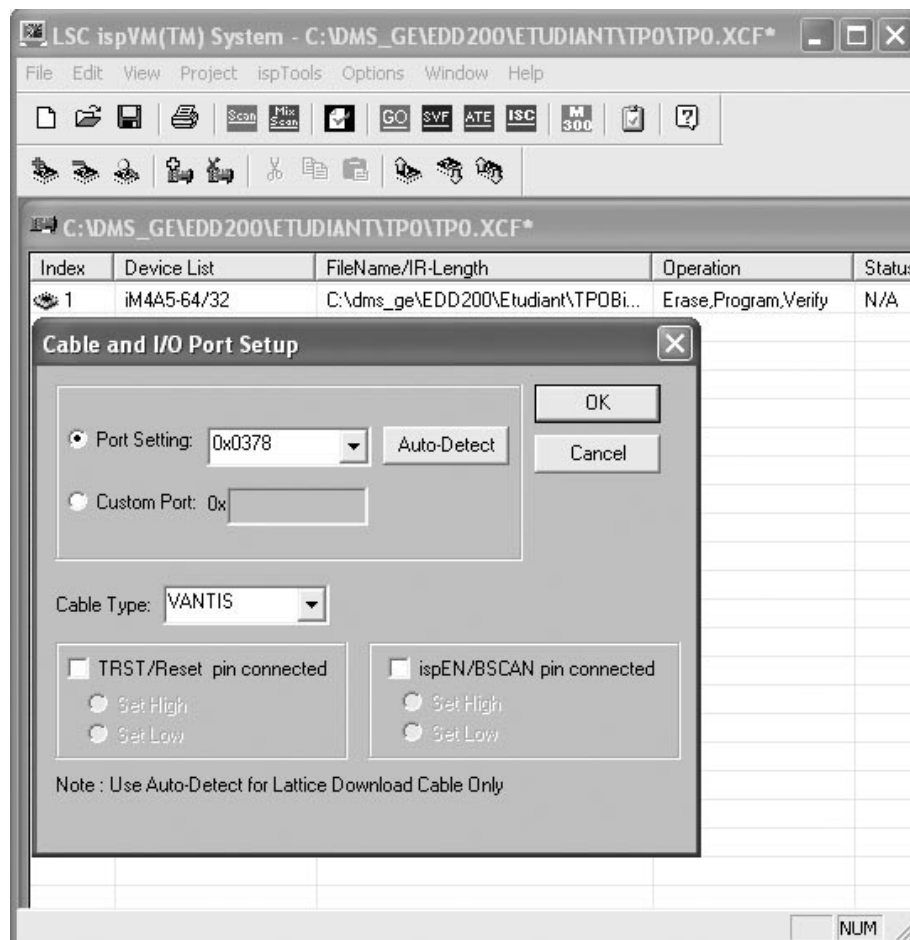
0.16.9 Cliquer sur « **Browse** » pour sélectionner le chemin d'accès du fichier jedec à télécharger « **TP0Bis.jed** ».

Paramétrage de la vitesse de transmission, cliquer sur « **Project** », puis « **Project Settings** », puis « **Advanced** », régler le diviseur d'horloge de transmission à 10 pour un calculateur standard, valider.



Ne pas oublier que 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, diminuer la vitesse de transmission en augmentant le chiffre diviseur, et refaire un test de transmission.

Choix du type de câble, cliquer sur « **Options** », « **Cable and I/O port set Up** », valider le câble **VANTIS**. cliquer sur « **OK** »,



Sauvegarder le fichier sur « **File** » > « **Save as** »,
Choisir le chemin « C:\DMS_GEVEDD200\ETUDIANT\TP0Bis. »

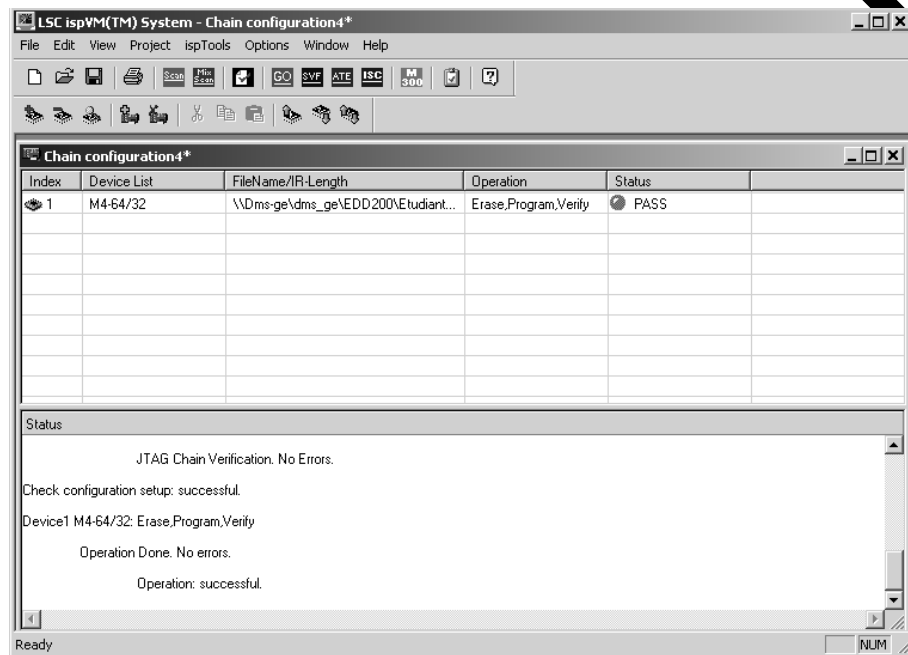
SPR

0.16.10 Téléchargement du fichier JEDEC.

Si le composant a déjà été utilisé, le logiciel pose la question 2 fois,
(XXXXX. Exist. Owerwrite ?), Répondre OK,

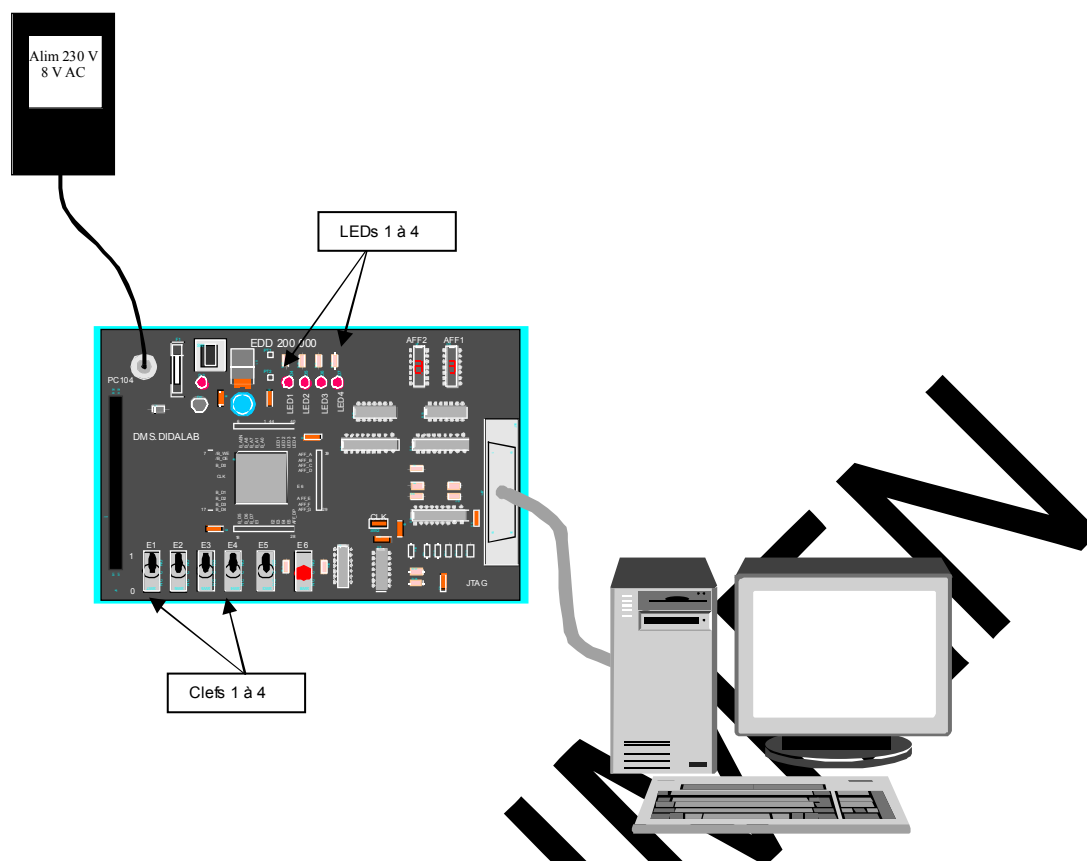
Cliquer ensuite sur « GO »,

Compte rendu de téléchargement :



Vérifier qu'aucun arrêt de téléchargement n'est signalée, si oui recommencer l'opération.

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



Vérifier le fonctionnement de interrupteurs comme variables d'entrés et les les en visualisation.

SPECIMEN

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.

1.2 FICHER VHDL

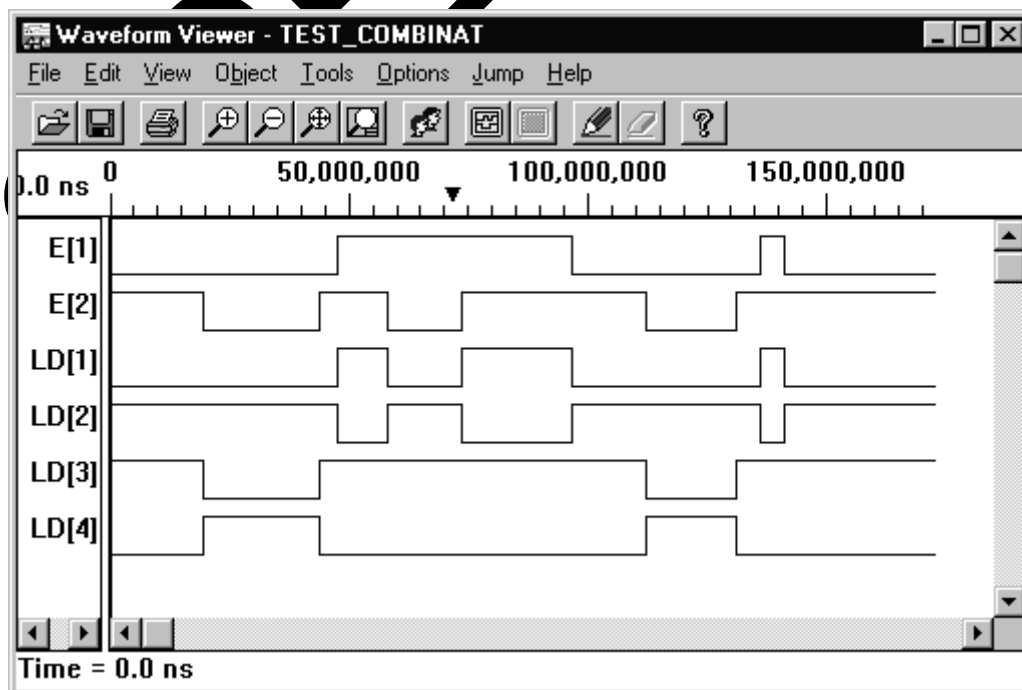
Le projet situé dans le sous-répertoire TP1 est : `TP1_COMBIN_simple.syn`

Le fichier VHDL : `COMBINAT.vhd` est donné ci-dessous.

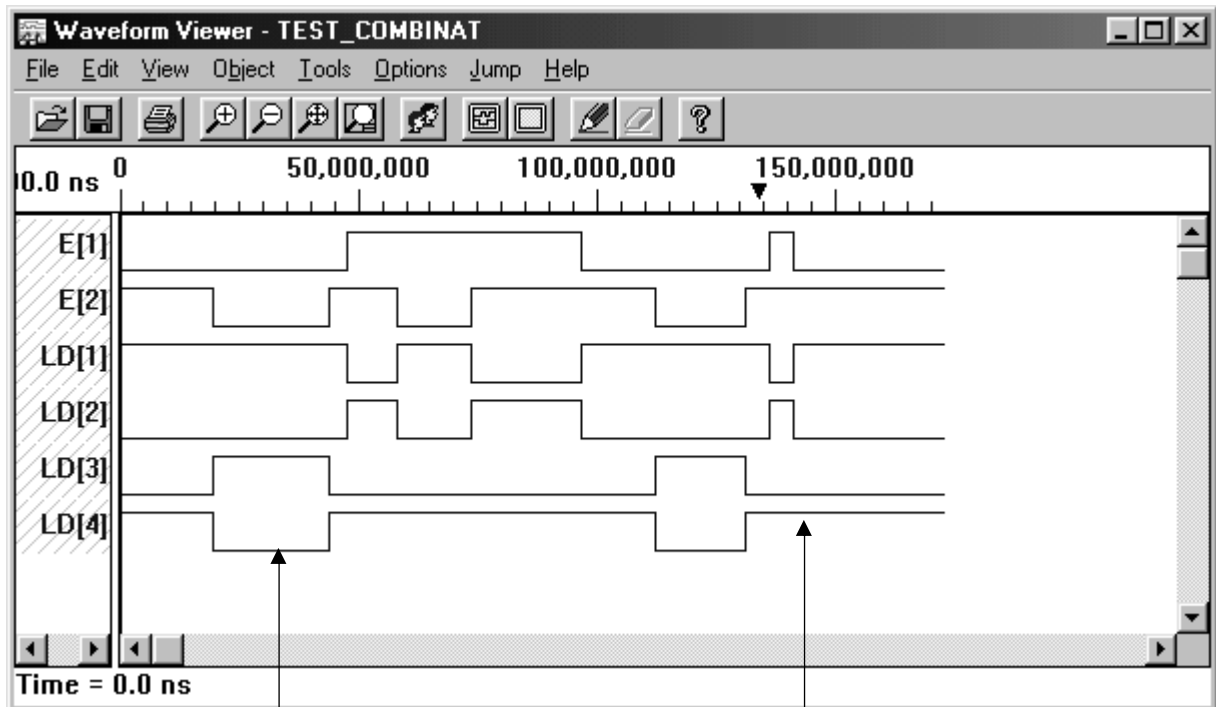
Un fichier de simulation `Test_combinat.vcd` réalisé grâce à un éditeur de stimuli de test permet de simuler les 4 fonctions.

Chronogrammes :

- Test des fonctions à l'aide d'un oscilloscope (fonctions vraies)



Test des fonctions avec les LEDs (fonctions complémentées).



LED4 allumée → niveau 1 LED4 éteinte → niveau 0
Fichier VHDL : ...\\TP1\\COMBINAT.VHD

```

-----
-- DESCRIPTION DES LIBRAIRIES
-----

library IEEE;
use ieee.std_logic_1164.all;

entity COMBINAT is
    PORT (
        E : in std_logic_vector (1 to 2); -- E1 et E2
        LD : out std_logic_vector (1 to 4) -- LED1 à LED4
    );
end;

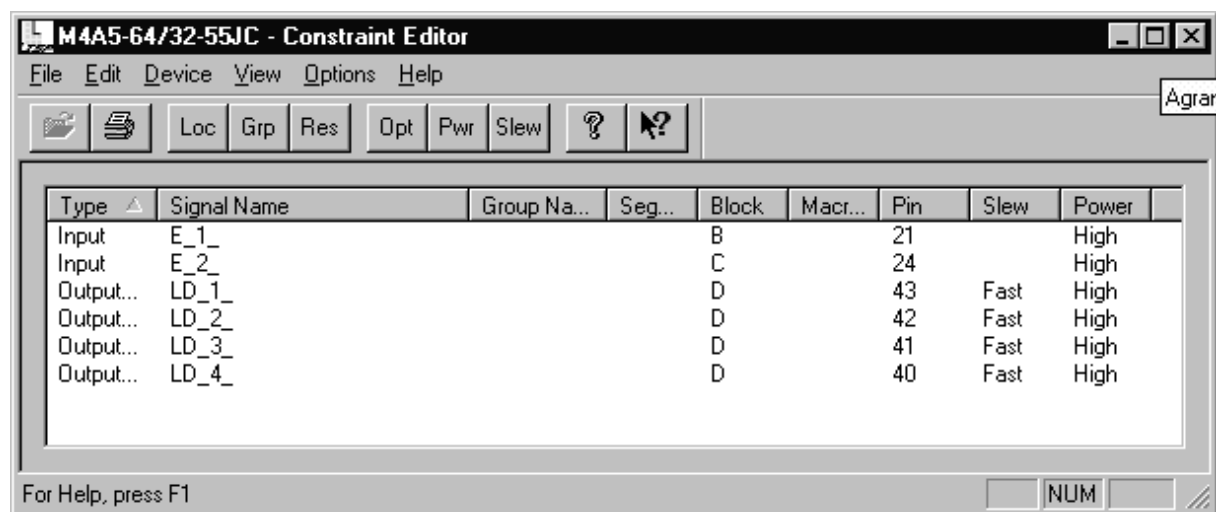
-----
-- DESCRIPTION DE L'ARCHITECTURE DES 4 FONCTIONS
-----

architecture PORTE of COMBINAT is
begin
    process(E)
    begin
        LD(1) <= not (E(1) and E(2));           -- AND
        LD(2) <= not (not (E(1) and E(2))); -- NAND
        LD(3) <= not (E(1) or E(2));          -- OR
        LD(4) <= not (not (E(1) or E(2))); -- NOR
    end process;
end architecture;
    
```

```
end PORTE;
```

1.3 AFFECTATION DES ENTREES / SORTIES

Les Entrées / Sorties sont assignées comme l'indique le tableau ci-dessous.



The screenshot shows the 'M4A5-64/32-55JC - Constraint Editor' window. The window has a menu bar (File, Edit, Device, View, Options, Help) and a toolbar with buttons for Loc, Grp, Res, Opt, Pwr, Slew, and help icons. A table lists the constraints for various signals. The table has columns for Type, Signal Name, Group Name, Segment, Block, Macro, Pin, Slew, and Power. The data rows are as follows:

Type	Signal Name	Group Na...	Seg...	Block	Macr...	Pin	Slew	Power
Input	E_1_			B		21		High
Input	E_2_			C		24		High
Output...	LD_1_			D		43	Fast	High
Output...	LD_2_			D		42	Fast	High
Output...	LD_3_			D		41	Fast	High
Output...	LD_4_			D		40	Fast	High

At the bottom of the window, there is a status bar with the text 'For Help, press F1' and a 'NUM' button.

SPECIMEN

SPECIMEN

TP 2 MULTIPLEXEUR

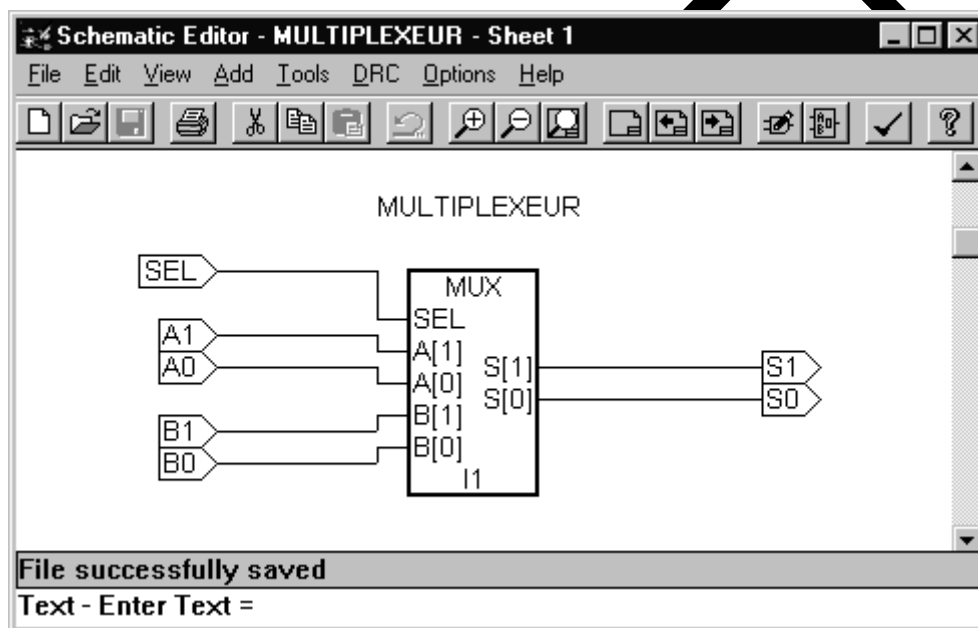
2.1 FONCTION MULTIPLEXEUR

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.



2.3 LE FICHIER VHDL

Le fichier VHDL correspondant décrit la table de vérité d'un multiplexeur.

Le projet situé dans le sous-répertoire TP2 est : `...\\TP2\\MULTIPLEX.syn`.

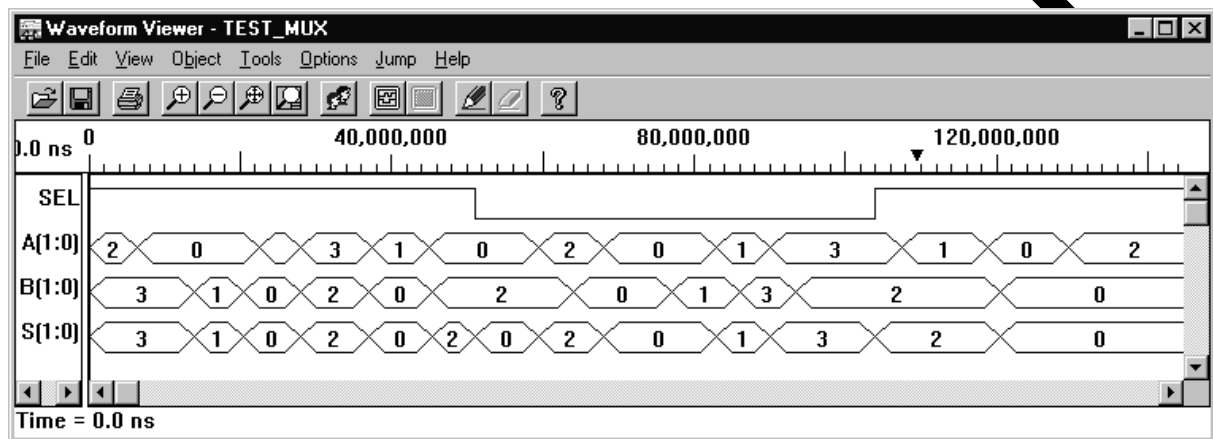
Fichier VHDL : c'est le fichier **MULTIPLEXEUR.vhd**

Il est donné ci-dessous.

Un fichier de simulation *Test_mux.wdl* réalisé grâce à un éditeur graphique de stimuli permet de simuler les fonctions multiplexeur.

- Pour la simulation, les instructions suivantes : **S <= not A** et **S <= not B** répondant aux soucis d'assimiler une LED allumée à un 1 ont été remplacées par : **S <= A** et **S <= B** ; cela permet d'observer directement la correspondance entre la sortie et l'entrée sélectionnée.

Chronogrammes



SPECIFIC

Fichier VHDL : ...TP2\ MULTIPLEX.VHD

```
-----  
--   DECLARATION DES LIBRAIRIES  
-----  
  
library ieee;  
use ieee.std_logic_1164.all;  
  
-----  
--   DECLARATION DE L'ENTITE MULTIPLEXEUR : MUX  
-----  
  
entity MUX is  
port ( SEL : in std_logic;                -- Sélction : E5  
      A  : in std_logic_vector (1 downto 0); -- E1 et E2  
      B  : in std_logic_vector (1 downto 0); -- E3 et E4  
      S  : out std_logic_vector (1 downto 0) -- LED1 et LED2  
      );  
end;  
  
-----  
--   ARCHITECTURE DU MULTIPLEXEUR : MUX  
-----  
  
architecture STRUCTURE of MUX is  
begin  
process (SEL, A, B)  
begin  
if SEL = '0' then  
    S <= not A;  
else  
    S <= not B;  
end if;  
end process;  
  
-- La complémentation "not" permet de faire correspondre une LED allumée à un niveau 1  
end STRUCTURE;
```

2.4 ASSIGNATION DES ENTREES / SORTIES

Elle est donnée par le tableau ci-dessous.

The screenshot shows a software window titled "M4A5-64/32-55JC - Constraint Editor". The window contains a menu bar (File, Edit, Device, View, Options, Help) and a toolbar with buttons for "Loc", "Grp", "Res", "Opt", "Pwr", "Slew", and help icons. Below the toolbar is a table with the following data:

Type	Signal Name	Group Na...	Seg...	Block	Macr...	Pin	Slew	Power
Input	A0			B		21		High
Input	A1			C		24		High
Input	B0			C		25		High
Input	B1			C		26		High
Input	SEL			C		27		High
Output/...	S0			D		43	Fast	High
Output/...	S1			D		42	Fast	High

At the bottom of the window, there is a status bar with the text "For Help, press F1" and a "NUM" button.

SPECIMEN

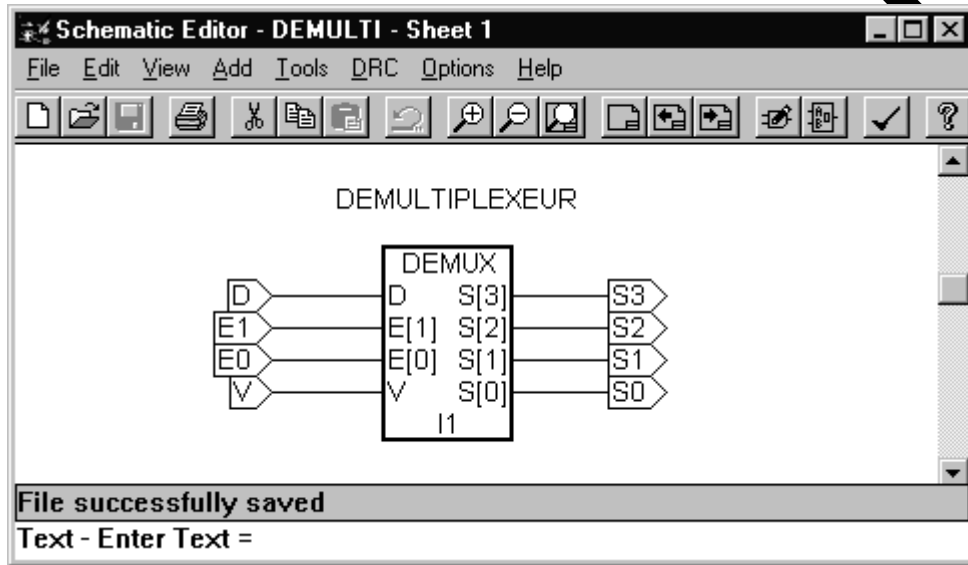
TP 3 DEMULTIPLEXEUR

3.1 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.



3.2 FICHIER VHDL

Le fichier VHDL décrit la table de vérité, figure ci-dessous d'un démultiplexeur 1 vers 4 avec une entrée de validation active à l'état 0 : toutes les sorties sont à 1 si $V = 1$.

Par ailleurs, les sorties sont actives à 0, donc $D = 1 \rightarrow S_i = 0$ si $V = 0_i$

E1	E0	V	S0	S1	S2	S3
X	X	1	1	1	1	1
0	0	0	\bar{D}	1	1	1
0	1	0	1	\bar{D}	1	1
1	0	0	1	1	\bar{D}	1
1	1	0	1	1	1	\bar{D}

Le projet situé dans le sous-répertoire TP3 est : **...TP3\DEMUX.syn** ; le fichier VHDL est : **DEMUL.VHD**.

Fichier VHDL : DEMUL.VHD

```

-----
--  DECLARATION DES LIBRAIRIES
-----

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

-----
--  DECLARATION DE L'ENTITE DEMULTIPEXEUR DEMUX
-----

entity DEMUX is
port ( D : in std_logic; -- E1, Données d'entrée
      E : in std_logic_vector(1 downto 0); -- E2, E3, Entrées de sélection
      V : in std_logic; -- E4, Entrée de validation active à 0
      S : out std_logic_vector(3 downto 0) -- Sorties de Données (LEA 0 à 3 à
); -- L[3:1]=S0) active à 1
end;

-----
--  DESCRIPTION DU DEMULTIPEXEUR
-----

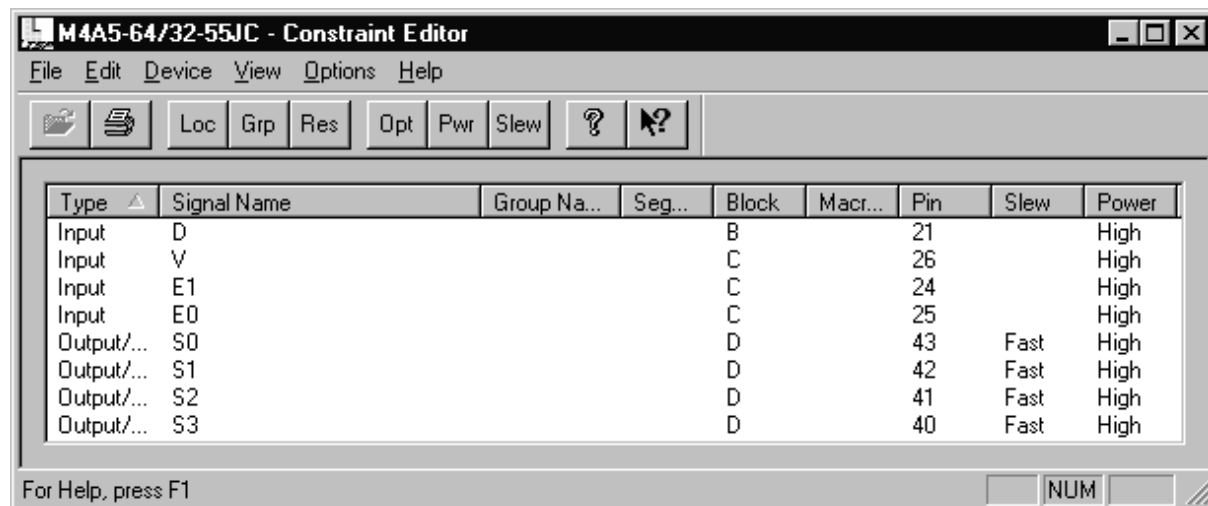
architecture DEMULTIPEX of DEMUX is
begin
S <= "1111" when V = '1' and not D = '1'
else "0111" when E = 0
else "1011" when E = 1
else "1101" when E = 2
else "1110" when E = 3
else "1111";

end DEMULTIPEX;

```

3.3 ASSIGNATION DES ENTREES / SORTIES

Elle est donnée par le tableau ci-dessous.



Type	Signal Name	Group Na...	Seg...	Block	Macr...	Pin	Slew	Power
Input	D			B		21		High
Input	V			C		26		High
Input	E1			C		24		High
Input	E0			C		25		High
Output/...	S0			D		43	Fast	High
Output/...	S1			D		42	Fast	High
Output/...	S2			D		41	Fast	High
Output/...	S3			D		40	Fast	High

SPECIMEN

SPECIMEN

TP 4 DECODEUR 7 SEGMENTS

4.1 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.

4.2 LE FICHER VHDL

Le fichier VHDL correspondant décrit la table de codage d'un afficheur segments.

Les segments sont allumés par des niveaux logiques 0.

Le projet situé dans le sous-répertoire TP4 est **TP4\SEPTSEGM.syn**.

Fichier VHDL : **...TP4\DECODESEG.VHD**.

```

-----
--DECLARATION DES BIBLIOTHEQUES
-----
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

-----
--DECLARATION DE L'ENTITE DECODEUR 7 SEGMENTS : SEG
-----
                                -- Entrées / Sorties
entity SEPTSEGM
port
    (      E   : in std_logic_vector (3 downto 0); -- E1 à E4
      CAFF : in std_logic;                       -- E5: Commande Affichage
      AFF1 : out std_logic;                       -- AFF1 : Affich. en BCD
      AFF2 : out std_logic;                       -- AFF2 : AAfich. en HEXA si E>9
      DP  : out std_logic;                       -- Decima Point
      SEG  : out std_logic_vector(6 downto 0) -- Segments g à a
    );
end;
```

Fichier VHDL : ...TP4\DECODSEG.VHD. suite

--DESCRIPTION DE L'ARCHITECTURE DU DECODEUR

```

architecture DECOD of SEG is
begin
process (E, CAFF)
begin
case E is
when "0000" => SEG <= "0000001"; --Ordre des SEG : abcdefg
when "0001" => SEG <= "1001111";
when "0010" => SEG <= "0010010";
when "0011" => SEG <= "0000110";
when "0100" => SEG <= "1001100";
when "0101" => SEG <= "0100100";
when "0110" => SEG <= "0100000";
when "0111" => SEG <= "0001111";
when "1000" => SEG <= "0000000";
when "1001" => SEG <= "0000100";

when "1010" => SEG <= "0000010"; --- E > "1001"
when "1011" => SEG <= "1100000";
when "1100" => SEG <= "0110001";
when "1101" => SEG <= "1000010";
when "1110" => SEG <= "0110000";
when "1111" => SEG <= "0111000";
when others => SEG <= "1111111";
end case;

if CAFF = '0' then
if (E > "1001") then AFF1 <= '1'; AFF2 <= '1';
else AFF1 <= '0'; AFF2 <= '1';
end if;
else AFF1 <= '1'; AFF2 <= '1';
end if;
DP <= CAFF;
end process;
end DECOD;

```

SPECIMEN

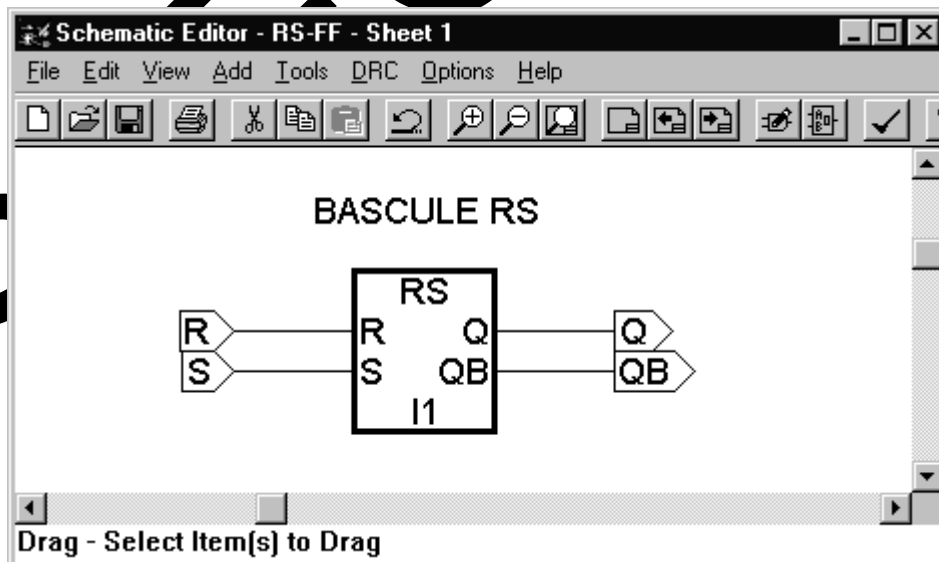
TP 5 BASCULE RS ELEMENTAIRE

5.1 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).



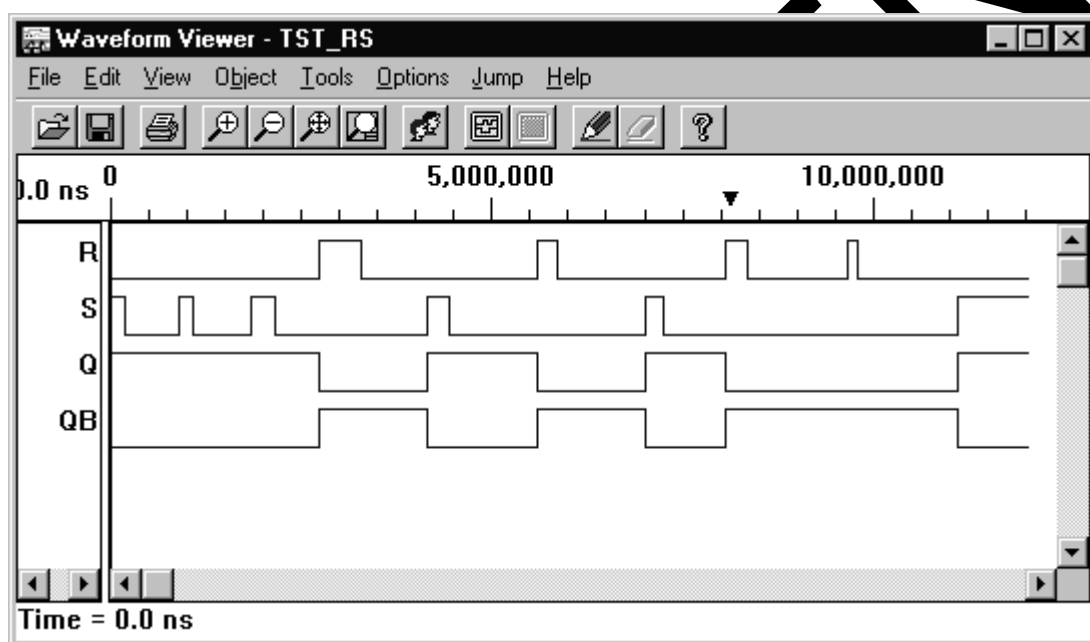
5.2 FICHER VHDL

Le fichier VHDL décrit la table de séquences ci-dessus.

Le projet situé dans le sous_ répertoire TP4 est : `... \TP5 \RS_FF.syn` ; le fichier VHDL est : ***RSFF.VHD***

Un fichier de test : `... \TP5 \TST_RS.WDL` permet de simuler la fonction RS avant le routage et la vérification in-situ.

Les chronogrammes sont les suivants :



Le fichier VHDL est donné ci-dessous.

On remarquera en particulier que les deux entrées R et S sont traitées séparément, servant de liste de sensibilité respectivement avec Q et QB dans deux processus distincts.

Fichier VHDL : ...\\TP5\\RSFF.VHDL

```
-----  
--      DECLARATION DES LIBRAIRIES  
-----  
library ieee;  
use ieee.std_logic_1164.all;  
  
-----  
--      DECLARATION DE L'ENTITE RS  
-----  
  
-- Entrées et Sorties  
entity RS is  
port ( R : in bit;      -- E1  
      S : in bit;      -- E2  
      Q : inout bit;   -- LED1  
      QB : inout bit  -- LED2  
      );  
end;  
  
-----  
--      DESCRIPTION DE LA BASCULE RS  
-----  
  
architecture BASCULE of RS is  
signal Y : std_logic;  
begin  
  
--      Description de la porte NOR ayant pour entrée R et QB  
  
process(R, QB)  
begin  
Q <= not (R or QB);  
end process;  
  
--      Description de la porte NOR ayant pour entrée S et Q  
  
process(S, Q)  
begin  
QB <= not (S or Q);  
end process;  
  
-- La combinaison R = S = 1 est évidemment interdite !!!  
  
end BASCULE;
```

BASCULE JK

5.3 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	$\overline{Q_n}$	Complémentation de la sortie

Le projet est de type VHDL seulement

5.4 LE FICHER VHDL

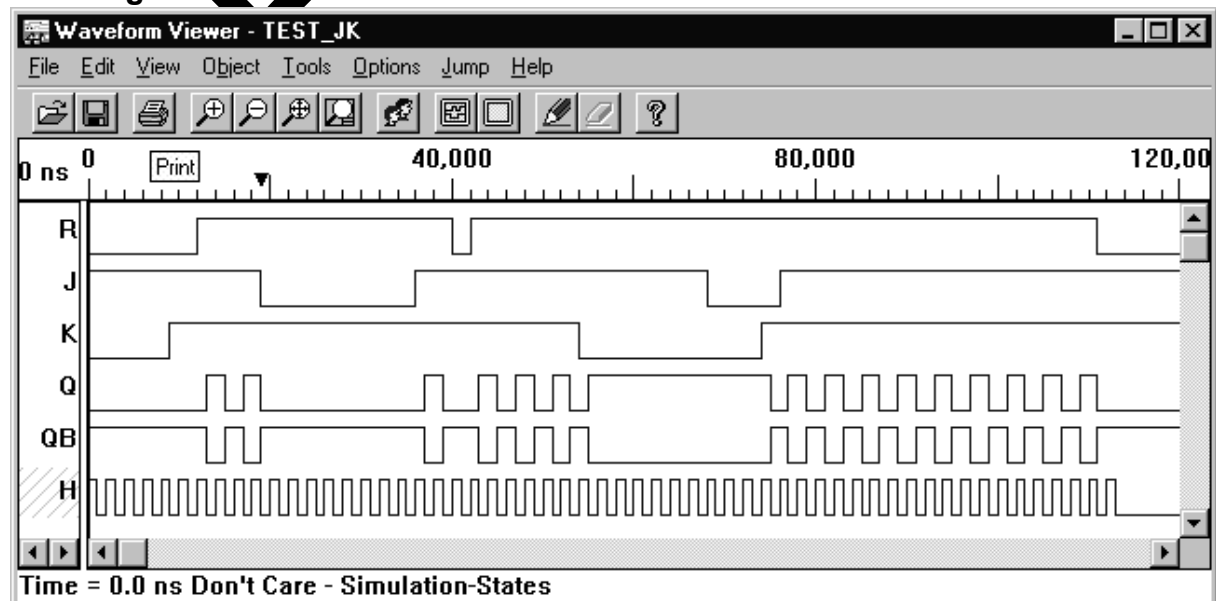
Le projet situé dans le sous-dépotoire TP est le suivant : ...**TP6\JKFF.syn**

Le fichier VHDL : **JK_FF.VHD** vérifie la table de vérité de la bascule JK ; la complémentation de sorties n'est pas nécessaire pour la correspondance entre niveau 1 et allumage des LEDs.

Un fichier de test **TEST_JK.WDL** permet de simuler la fonction avant routage. Les chronogrammes sont donnés ci-dessous.

Vérifier que la complémentation de sorties se fait bien sur le front descendant.

Chronogrammes



Fichier VHDL : ...\\TP6\\JK_FF.VHD

```

-----
--      DECLARATION DES LIBRAIRIES
-----

library ieee;
use ieee.std_logic_1164.all;

-----
--      DECLARATION DE L'ENTITE JK
-----

                                -- Entrées et Sorties
entity JK is
port ( R : IN std_logic; -- E1
      J : IN std_logic; -- E2
      K : IN std_logic; -- E3
      H : IN std_logic; -- E6
      Q : out std_logic;  -- LED1
      QB :out std_logic  -- LED2 (Q barre)
);
end;

-----
--      DESCRIPTION DE LA BASCULE JK
-----

architecture BJK of JK is
signal X : std_logic;  -- signal interne

begin
process ( R, J, K, H )
begin
if (R = '0') then X = '0';
elsif (H = '0' and 'event H) then -- Horloge active sur front descendant

-- TABLE DE VERITE DE LA BASCULE JK :
      (J = '1' and K = '0') then X <= '1';
      elsif (J = '0' and K = '1') then X <= '0';
      elsif (J = '1' and K = '1') then X <= not X;
      else X <= X; -- J = K = 0
      end if;
end process;

-----
--      AFFECTATION DU SIGNAL X A Q ET QB
-----

Q <= X;
QB <= not X;

end BJK;

```

SPECIMEN

TP 6 COMPT. / DECOMPT. BCD et PREDIV. PROGRAMMABLE

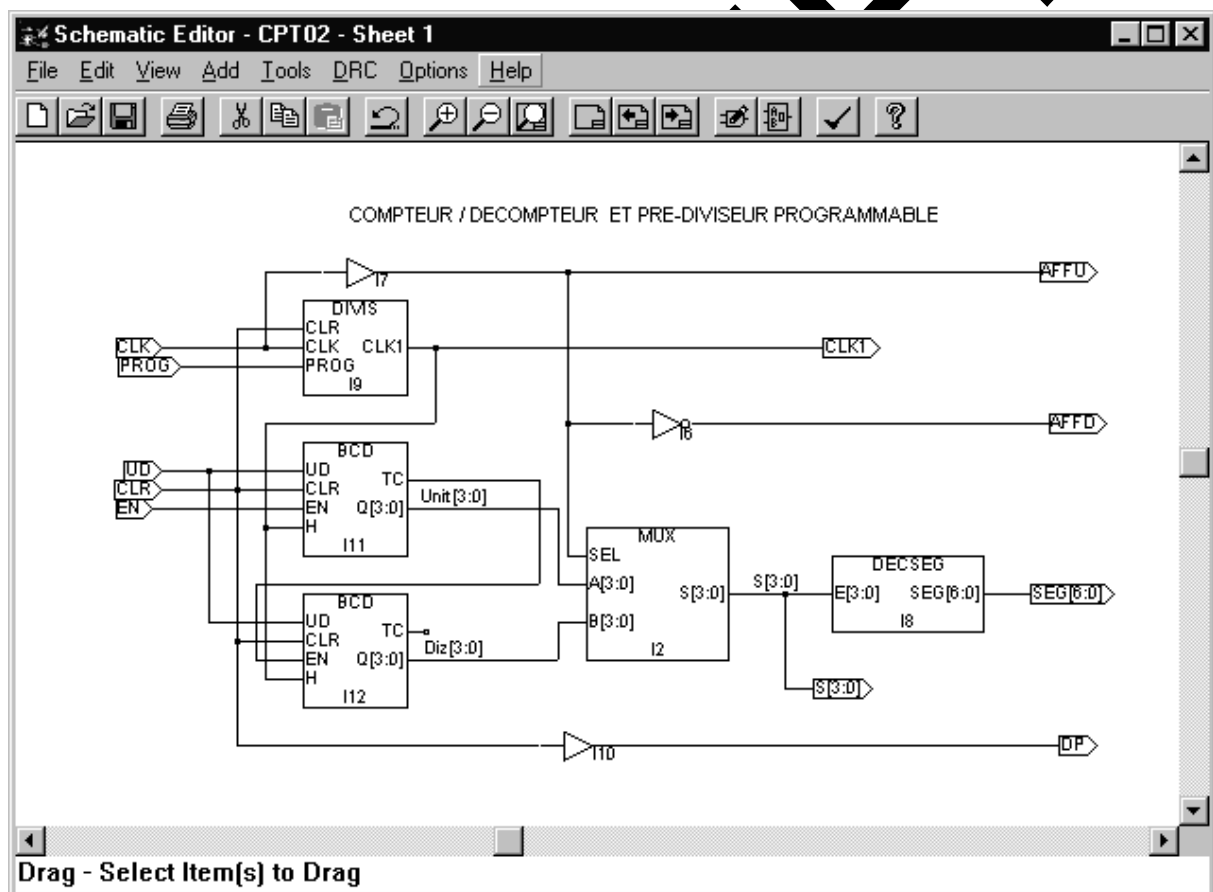
6.1 COMPTEUR / DECOMPTEUR – PREDIVISEUR PROGRAMMABLE

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.

6.2 LES MODULES

Le schéma est donné ci-dessous.



6.2.1 PREDIVISEUR

Pour des raisons de lisibilité, l'horloge utilisée est de 2 Hz.

Le prédiviseur programmable fournit au compteur / décompteur un signal d'horloge CLK1 de fréquence programmable selon l'entrée PROG (E6) et les nombres définis dans le fichier *DIV.VHDL*.

6.2.2 COMPTEUR / DECOMPTEUR BCD

En fonction de l'entrée UD (E3), le compteur compte de 0 à 9, ou décompte de 9 à 0 si l'entrée d'autorisation EN (E2) est à 1.

6.2.3 LE MULTIPLEXEUR

Il permet de présenter au décodeur 7 segments, les unités ou les dizaines sortant des compteurs.

6.2.4 DECODEUR 7 SEGMENTS

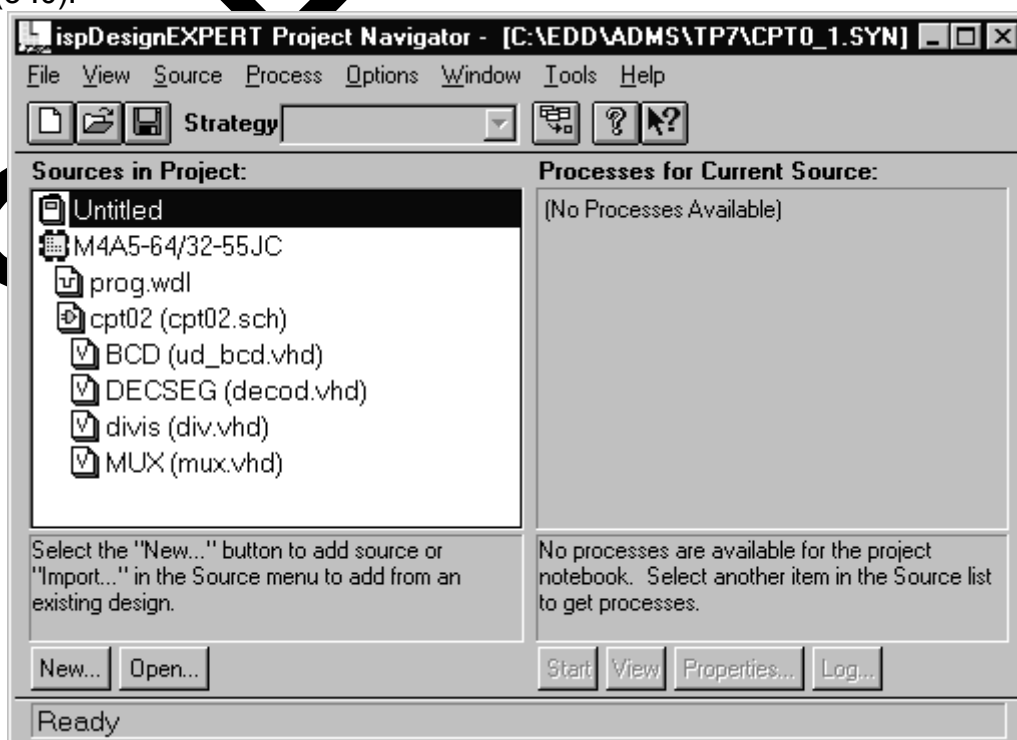
Il fournit aux afficheurs, les états logiques correspondant au chiffre à afficher. Les signaux de commande des afficheurs sont réalisés directement dans le schéma.

Les sorties CLK1 et S(3 :0) sur le schéma servent éventuellement pour la simulation, mais ne sont pas accessibles à l'extérieur.

6.3 LES FICHIERS

Le projet est dans le sous-répertoire ...TP7 : ...TP7\CPT0_1.syn.

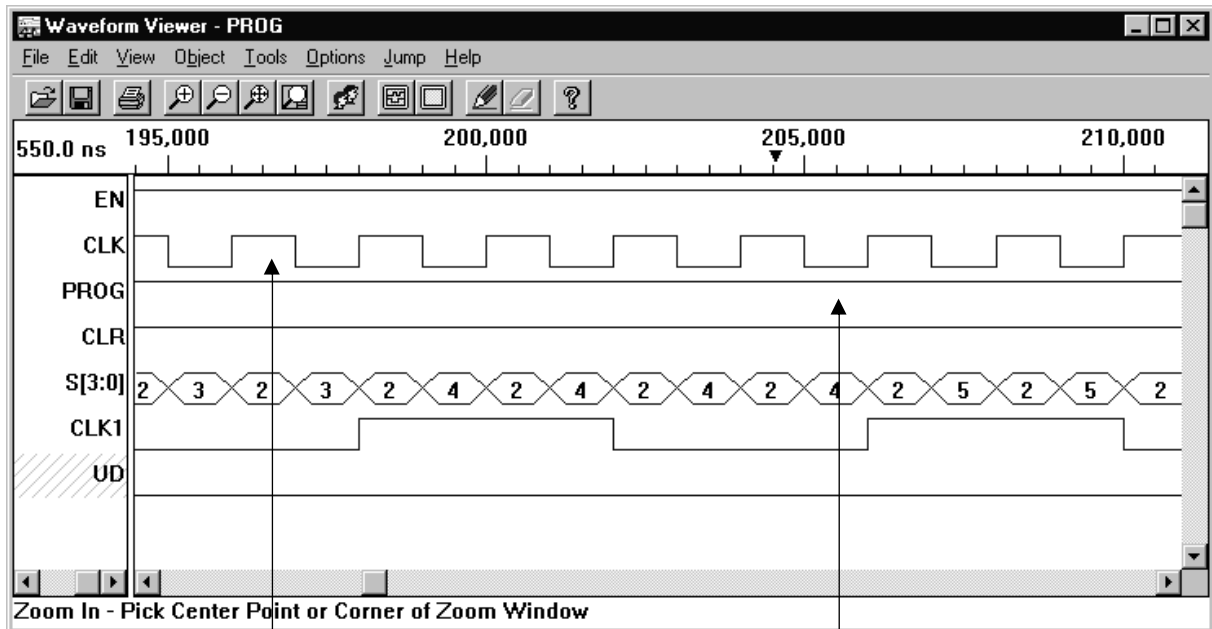
Dans le même sous-répertoire nous avons les fichiers VHDL suivant, ainsi qu'un fichier test : **prog.wdl** permettant de visualiser le signal CLK1 et les données S(3 :0).



6.3.1 CHRONOGRAMMES

Les chronogrammes suivants correspondent au même rapport de division : $N1 = 4$, et met en évidence le comptage et le décomptage.

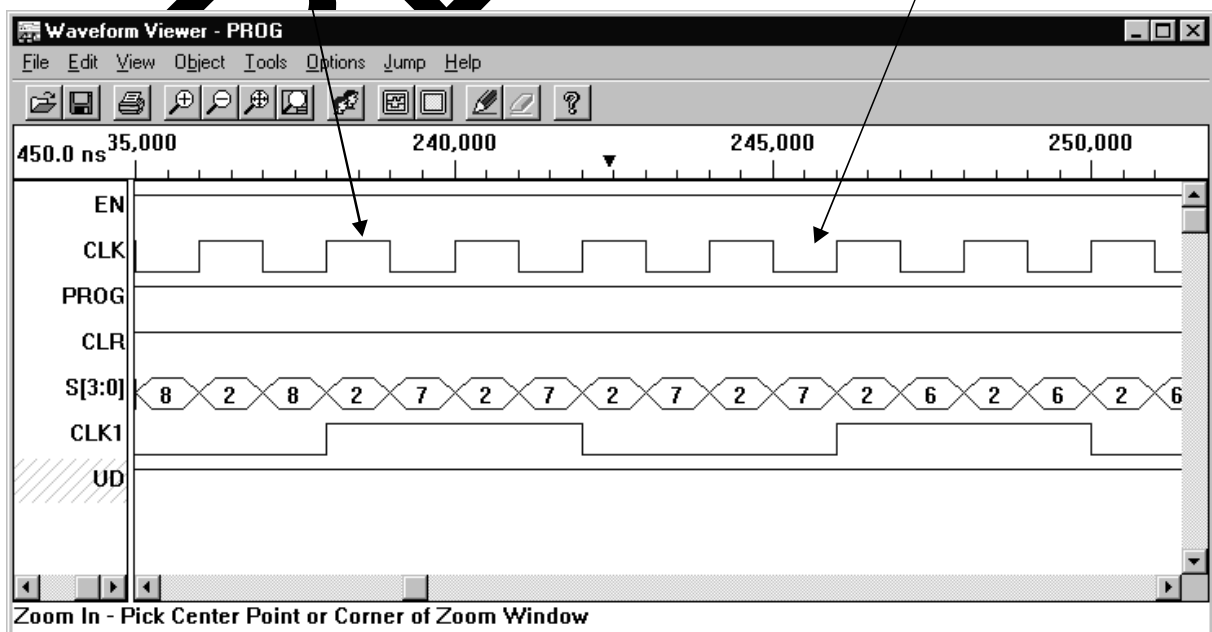
Comptage : $PROG = 1$ $UD = 0$



CLK = 1 → S = Sortie Dizaines

CLK = 0 → S = Sortie Unités

Décomptage : $PROG = 0$ $UD = 1$



Fichier : UD_BCD.VHD Module Compteur / Décompteur

```

-----
--      DECLARATION DES LIBRAIRIES
-----

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

-----
--      DCLARATION DE L'ENTITE DECAD
-----

entity BCD is
port
  ( UD : IN std_logic;      -- Sens de comptage/décomptage
    CLR : IN std_logic;    -- Remise à Zéro
    EN : IN std_logic;     -- Autorisation de compter
    H  : IN std_logic;     -- Horloge
    TC : OUT std_logic;    -- Terminal Count
    Q  : OUT std_logic_vector(3 downto 0) -- Sorties compteur
  );
end entity;

-----
--      ARCHITECTURE DU COMPTEUR DECOMPTEUR
-----

architecture COMPTEUR of BCD is
--      Définition des signaux internes
signal X : std_logic_vector(3 downto 0);
signal Y : std_logic;
signal Z : std_logic;

begin
  process (CLR, EN, H, UD)
-----
--      Remise à Zéro
-----
begin
  if (CLR = '0') then
    X <= x"0000";    Y <= '0';    Z <= '0';

    elsif ((H = '1' and H'event) and EN = '1') then
-----
--      Comptage
-----

```

```
if UD = '0' then
  if X = x"9" then X <= x"0";
  else X<= X + 1;
  end if;
-----
--          Décomptage
-----

  elsif X = x"0" then X <= x"9";
  else X <= X - 1;

end if;
end if;
end process;
-----
--          Génération Fin comptage / décomptage
-- Affection du vecteur signal interne au vecteur sortie
-----

--          Fin de comptage / décomptage

process (UD,X)
begin
if UD = '0' and X = 9 then TC <= '1'; --else TC <= '0';
  elsif UD = '1' and X = 0 then TC <= '1'; else TC <= '0';
end if;
end process;

--          Affection

Q <= X;

end COMPTEUR;
```

SPECIMEN

Fichier : DECOD.VHD Module DECODEUR 7 SEGMENTS

```

-----
--   DECLARATION DES LIBRAIRIES
-----

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

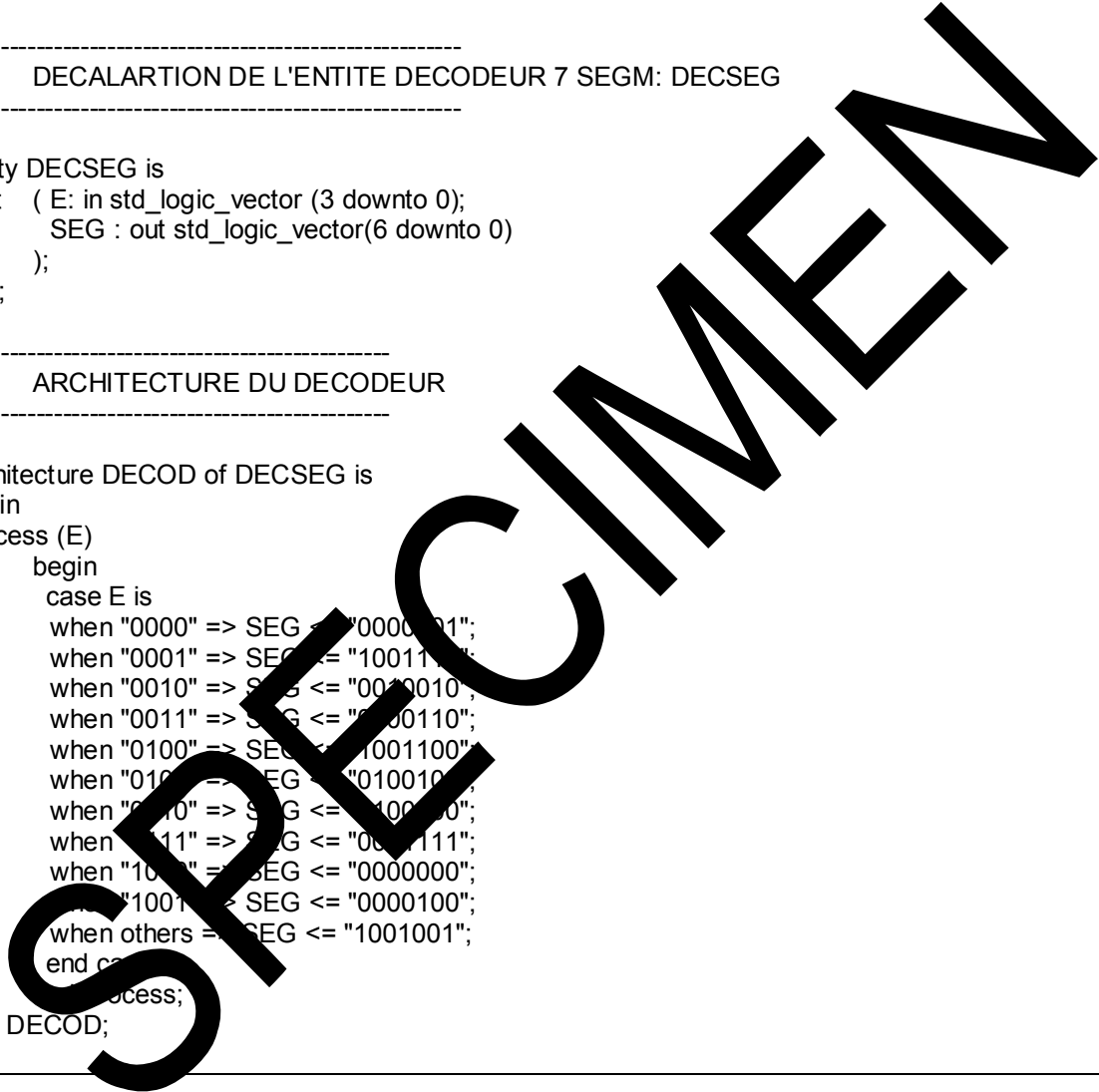
-----
--   DECLARATION DE L'ENTITE DECODEUR 7 SEGM: DECSEG
-----

entity DECSEG is
port  ( E: in std_logic_vector (3 downto 0);
        SEG : out std_logic_vector(6 downto 0)
        );
end;

-----
--   ARCHITECTURE DU DECODEUR
-----

architecture DECOD of DECSEG is
begin
process (E)
begin
  case E is
    when "0000" => SEG <= "0000001";
    when "0001" => SEG <= "1001111";
    when "0010" => SEG <= "0010010";
    when "0011" => SEG <= "1001110";
    when "0100" => SEG <= "1001100";
    when "0101" => SEG <= "0100100";
    when "0110" => SEG <= "1000000";
    when "0111" => SEG <= "0011111";
    when "1000" => SEG <= "0000000";
    when "1001" => SEG <= "0000100";
    when others => SEG <= "1001001";
  end case;
end process;
end DECOD;

```



Fichier : *DIV.VHD*Module **PREDIVISEUR PROGRAMMABLE**

```

-----
--   DECALARTION DES LIBRAIRIES
-----

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

-----
--   DECALARTION DE L'ENTITE DIVISEUR: DIV
-----
entity divis is
port  ( CLR : in std_logic;
        CLK : in std_logic;
        PROG : in std_logic;
        CLK1 : out std_logic
        );
end;

-----
-- DECALARTION DE L'ARCHITECTURE DU DIVISEUR PAR 100: DIV
-----

architecture DIV100 of DIVIS is
signal X : std_logic_vector(0 to 7); -- signal interne
signal Y : bit; -- Signal interne

begin
process(CLR, PROG, CLK)
begin
if CLR = '0' then
X <= x"00";
elsif (CLK = '1' and CLK'event) then
if ((PROG = '0') and (X >= x"2")) then
X <= x"00";
elsif ((PROG = '1') and (X >= x"1")) then
X <= x"00";
else X <= X + 1;
end if;
end if;
end process;

Y <= '1' when X <= x"00" else '0';

-----
-- GENERATION D'UN SIGNAL D'HORLOGE SYMETRIQUE
-----

process(Y)
begin

```

```
if (Y = '1' and Y'event) then
    CLK1 <= not CLK1;
else CLK1 <= CLK1;
end if;
end process;

end DIV100;
```

SPECIMEN

Fichier : **MUX.VHD**Module **MULTIPLEXEUR**

```
-----  
--   DECALARTION DES LIBRAIRIES  
-----  
  
library ieee;  
use ieee.std_logic_1164.all;  
  
-----  
--   DECALARTION DE L'ENTITE  
-----  
  
entity MUX is  
  
port  (SEL : in std_logic;  
       A, B : in std_logic_vector(3 downto 0);  
       S   : out std_logic_vector(3 downto 0));  
end;  
  
-----  
--   ARCHITECTURE DE L'ENTITE MULTIPLEXEUR  
-----  
  
architecture MULTIPLEX of MUX is  
begin  
process (A, B, SEL)  
begin  
if SEL = '0' then  
    S <= A;  
else  S <= B;  
end if;  
end process;  
end MULTIPLEX
```

SPECIMEN

ASSIGNATION DES ENTREES / SORTIES

M4A5-64/32-55JC - Constraint Editor

File Edit Device View Options Help

Loc Grp Res Opt Pwr Slew ? ?

Type	Signal Name	Group Name	Segm...	Block	Macr...	Pin	Slew	Power
Input	CLR			B		21		High
Input	EN			C		24		High
Input	CLK					11		
Input	UD			C		25		High
Input	PROG					33		
Output/...	CLK1			D		40	Fast	High
Output/...	DP			C		28	Fast	High
Output/...	SEG_0_			C		29	Fast	High
Output/...	SEG_1_			C		30	Fast	High
Output/...	SEG_2_			C		31	Fast	High
Output/...	SEG_3_			D		36	Fast	High
Output/...	SEG_4_			D		37	Fast	High
Output/...	SEG_5_			D		38	Fast	High
Output/...	SEG_6_			D		39	Fast	High
Output/...	AFFU			D		43	Fast	High
Output/...	AFFD			D		42	Fast	High

For Help, press F1

NUM

SPECIMEN