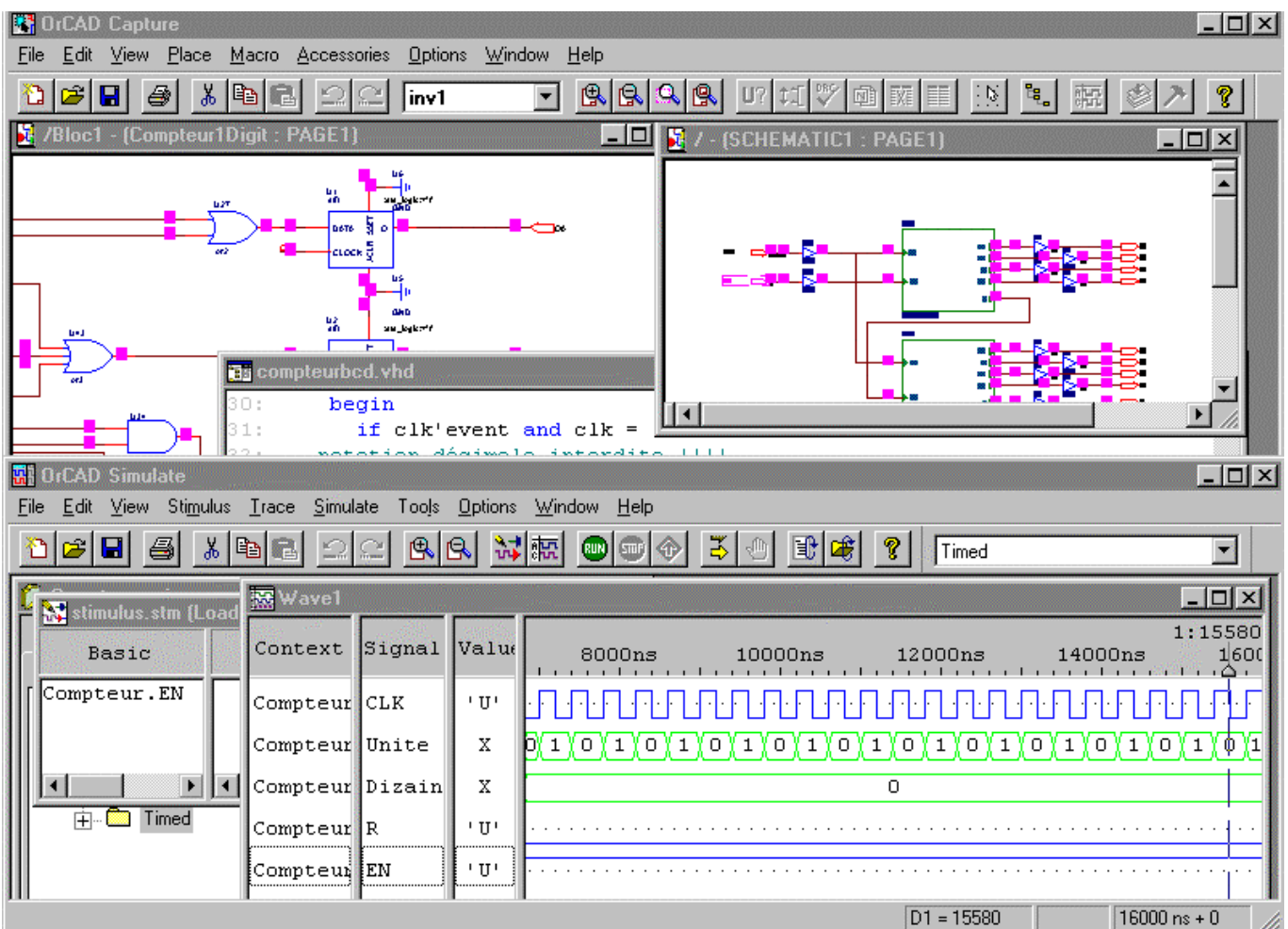


OrCad Express 9.1 Circuits Logiques Programmables




<p>ORCAD EXPRESS</p> <p>DEVELOPPEMENT DE CIRCUITS LOGIQUES PROGRAMMABLES</p>
--

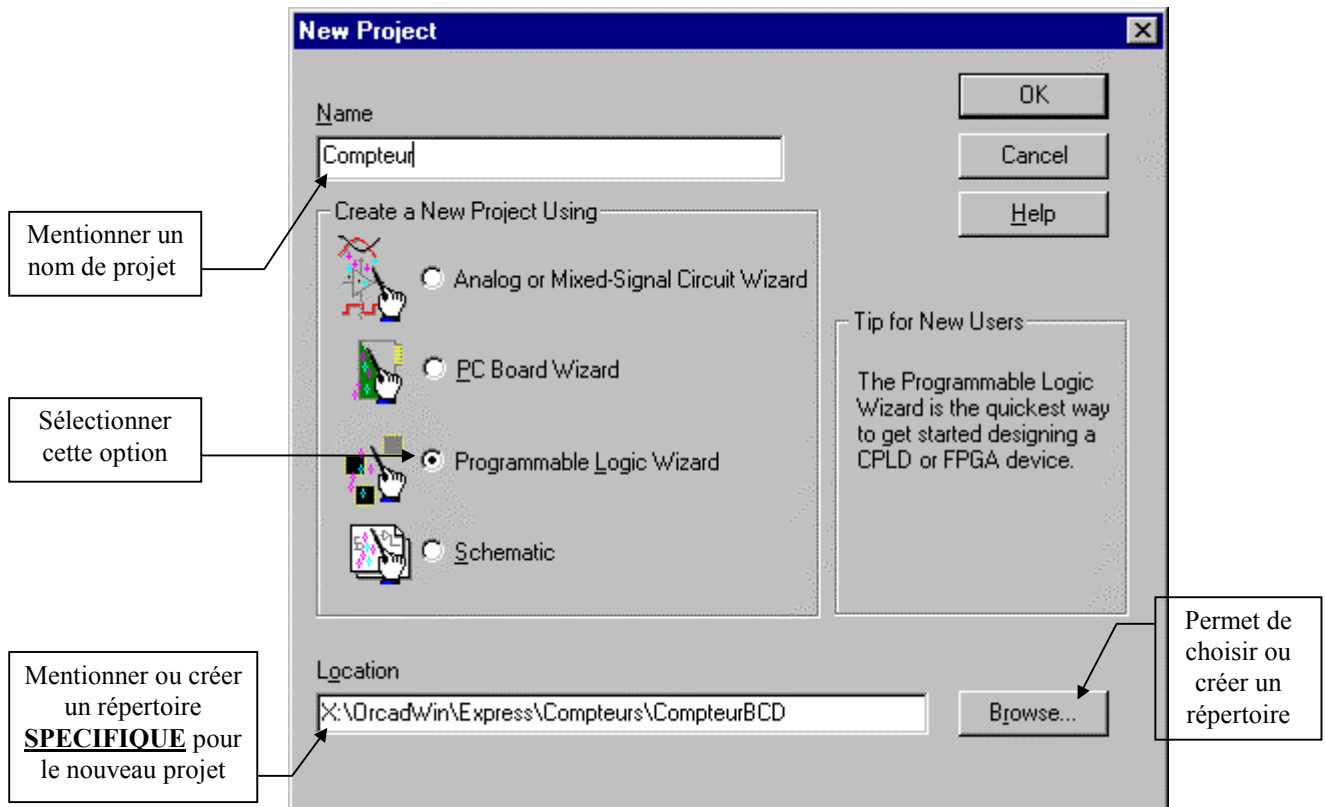
1) Création d'un projet	3
1.1) Définition du projet.....	3
1.2) Sélection de la famille de circuits logiques programmables.....	4
1.3) Description de la fonction réalisée par le circuit logique programmable	4
1.3.1) Description schématique.....	4
1.3.2) Description mixte : schéma / VHDL	5
1.3.3) Description en langage VHDL seulement.	6
1.4) Représentation hiérarchique d'un projet.....	7
1.4.1) Définition d'un bloc hiérarchique associé à une représentation schématique.....	7
1.4.2) Définition d'un bloc hiérarchique associé à une description en langage VHDL	10
1.5) Principaux symboles de base (Librairie SPLD).....	11
1.6) Contraintes relatives au brochage des circuits logiques programmables SPLD.....	11
1.7) Contraintes relatives à la description de projet sous forme schématique	13
1.8) Exemples de projets destinés à des circuits logiques programmables SPLD	15
1.8.1) Exemple de projet « Schéma simple ».....	15
1.8.2) Exemple de projet « Schéma hiérarchique »	16
1.8.3) Exemple de projet « Schéma hiérarchique et code VHDL ».....	17
1.8.4) Exemple de projet « Code VHDL seul ».....	18
2) Compilation du projet.....	19
2.1) Vérification syntaxique (du langage VHDL).....	19
2.2) Compilation du projet	19
2.3) « Routage » du circuit	21
3) Simulation.....	23
3.1) Lancement du simulateur	23
3.2) Définition des signaux d'entrée	24
3.2.1) Editeur de stimulus	24
3.2.2) Groupes de signaux - Bus	28
3.2.3) Sauvegarde des stimulus.....	29
3.3) Simulation du projet.....	30
3.3.1) Lancement - Durée de simulation.....	30
3.3.2) Choix des signaux à visualiser.....	30
3.3.3) Groupes de signaux - Bus	31
3.3.4) Mesures de durées - Curseurs	33
3.4) Modification du projet	33
4) Quitter le simulateur.....	33
5) Impressions : Schémas - Code VHDL - Résultats de simulation.....	33


1) Création d'un projet

1.1) Définition du projet

Pour pouvoir réaliser la programmation d'un circuit logique programmable (Programmable Logic Device : PLD), il est nécessaire de créer un projet spécifique. A partir d'OrCad Capture, créer un nouveau projet en choisissant

l'option du menu File - New - Project ou en cliquant sur l'icône  (Create Document).

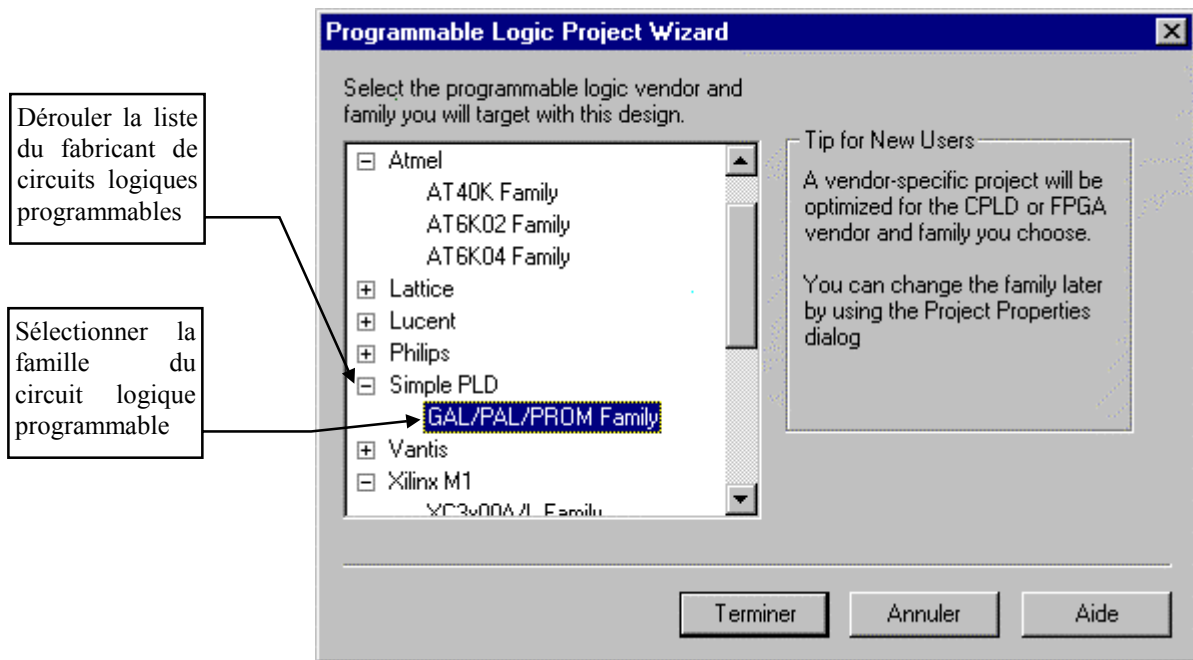


 Le nom donné au projet ne doit comporter NI SIGNES DE PONCTUATION, NI ESPACE, NI LETTRES ACCENTUEES.

Certains outils logiciels fournis par les fabricants de circuits logiques programmables n'acceptent que des noms ayant 8 caractères au maximum.

1.2) Sélection de la famille de circuits logiques programmables

La boîte de dialogue suivante permet de choisir le fabricant et la famille de circuits logiques programmables.



1.3) Description de la fonction réalisée par le circuit logique programmable

La fonction réalisée par le circuit logique programmable peut être décrite par :

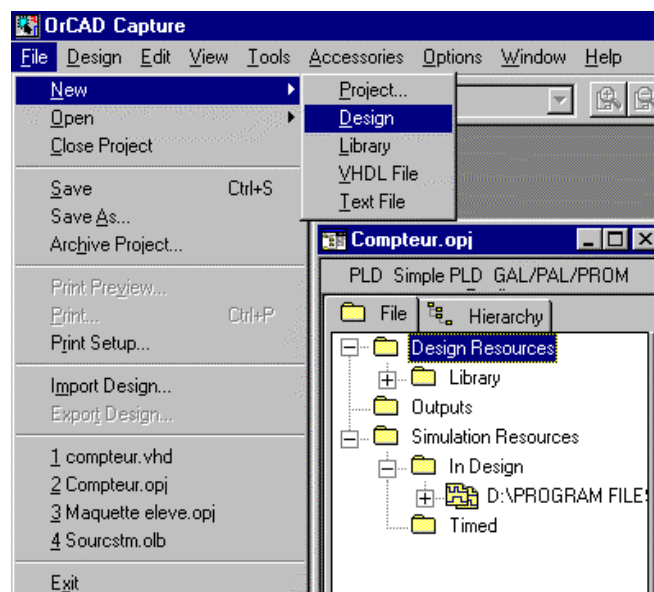
- une description schématique (voir § 1.3.1);
- une description mixte : schéma / VHDL (voir § 1.3.2) ;
- une description comportementale en langage VHDL (voir § 1.3.3).

Les descriptions peuvent éventuellement reposer sur une représentation hiérarchique décrite au § 1.4).


L'étape suivante consiste donc à définir le mode de description choisi en ajoutant dans le gestionnaire de projets le ou les éléments correspondants.

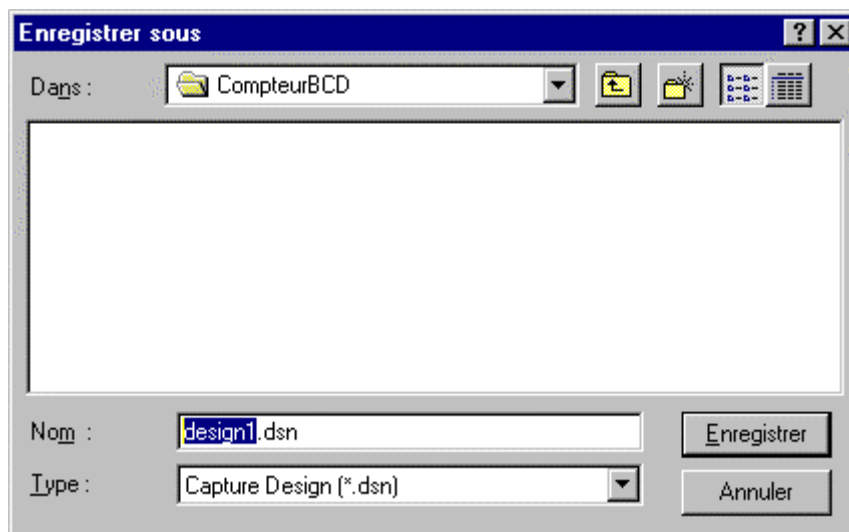
1.3.1) Description schématique

Le gestionnaire de projets ne fait référence à aucun schéma. Pour en créer un nouveau, choisir l'option du menu File - New - Design.

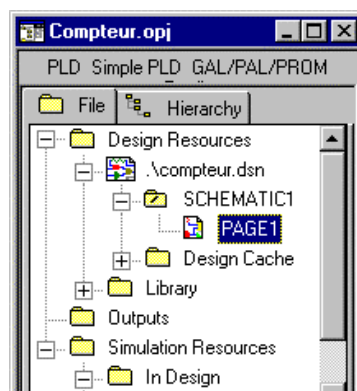


Un nouveau schéma est maintenant disponible.

- **Enregistrer dès maintenant** le schéma en choisissant l'option du menu File - Save ou en cliquant sur l'icône  (Save Document).
- **Attention** : le répertoire par défaut dans lequel sera enregistré le fichier schéma n'est pas le répertoire du projet. Il est indispensable de parcourir l'arborescence pour retrouver le répertoire contenant le projet.
- Le nom par défaut « design1.dsn » doit absolument être remplacé par le même nom que le projet (Compteur.dsn dans l'exemple), en mentionnant l'extension .DSN.



Le schéma est maintenant disponible dans le gestionnaire de projets.

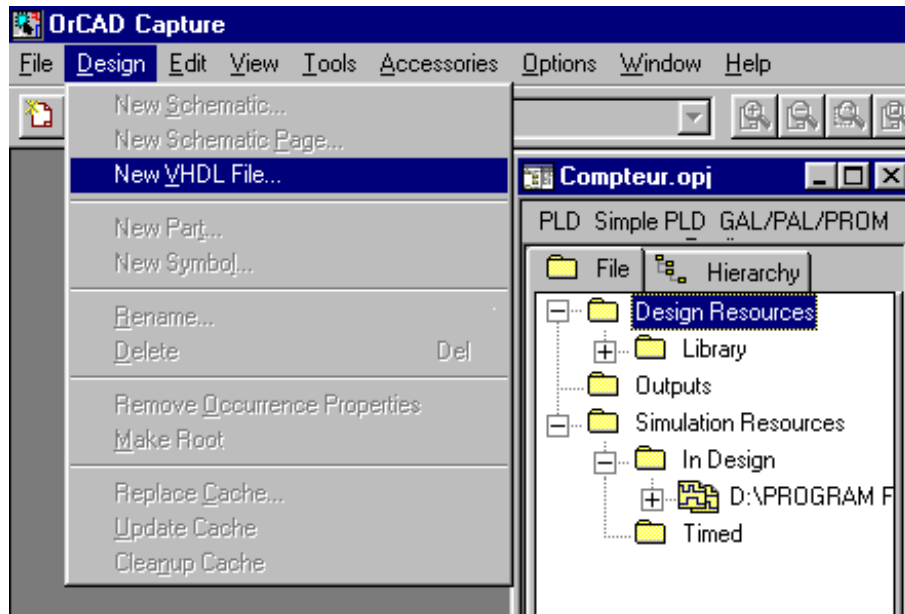



1.3.2) Description mixte : schéma / VHDL

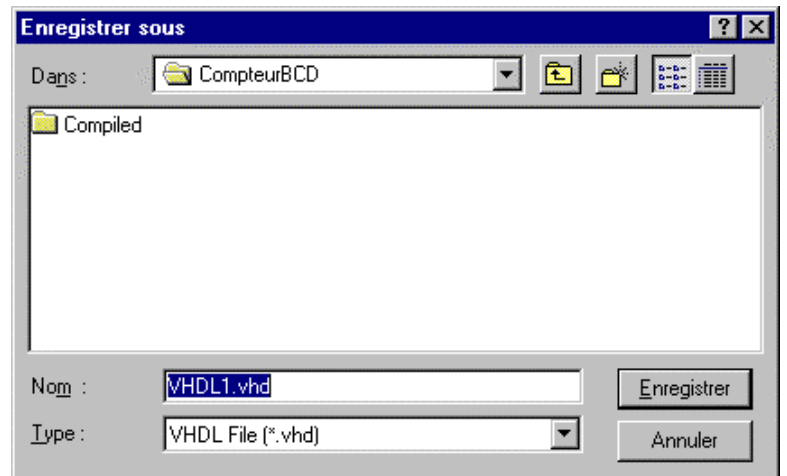
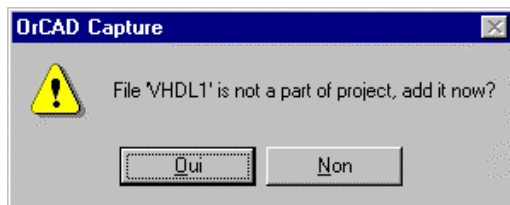
Un projet peut être constitué de représentations schématiques associées à une description en langage VHDL. La création du projet nécessite tout d'abord l'ajout d'un schéma (voir § 1.3.1). Le ou les fichiers VHDL seront ajoutés en suivant la méthode décrite dans le §1.4.2.

1.3.3) Description en langage VHDL seulement.

Le gestionnaire de projets ne fait référence à aucun fichier source en VHDL. Pour en créer un nouveau, choisir l'option du menu Design - New VHDL File.

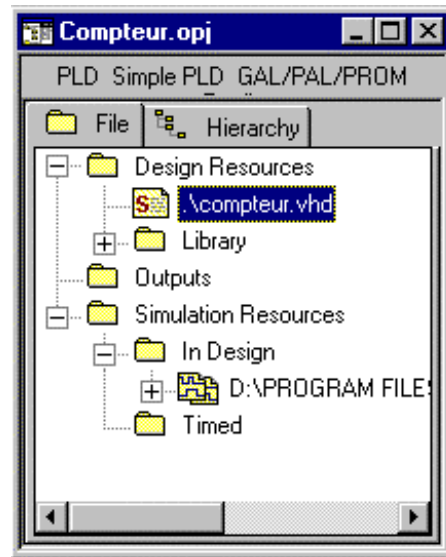


Un nouveau fichier est créé. Son nom par défaut est « VHDL1 ». Enregistrer dès maintenant le fichier source en cliquant sur l'icône  puis, répondre « Oui » à la question suivante.



 **Le nom par défaut « VHDL1.vhd » doit absolument être remplacé par le même nom que le projet (Compteur.vhd dans l'exemple), en mentionnant l'extension .VHD.**

Le gestionnaire de projets contient maintenant le nom du fichier source VHDL. Un double clic sur son nom dans le gestionnaire de projets permet de l'éditer.




1.4) Représentation hiérarchique d'un projet

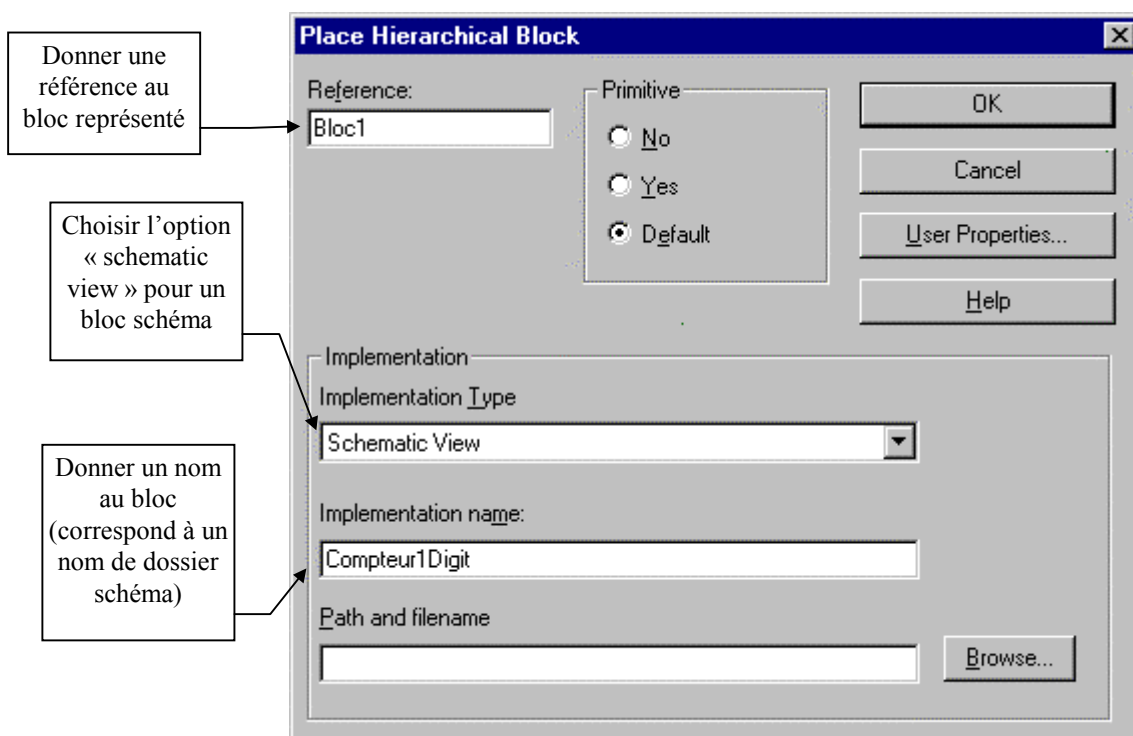
Dans le cas d'un projet complexe, il est souhaitable de décomposer les différentes fonctions réalisées par le circuit logique programmable en autant de schémas et/ou fichiers VHDL distincts. Il s'agit d'un projet hiérarchique.

L'organisation hiérarchique d'un projet fait appel à :

- des blocs hiérarchiques ;
- des broches hiérarchiques situées sur la périphérie d'un bloc ;
- des ports hiérarchiques situés en entrée et sortie des schémas internes.


1.4.1) Définition d'un bloc hiérarchique associé à une représentation schématique

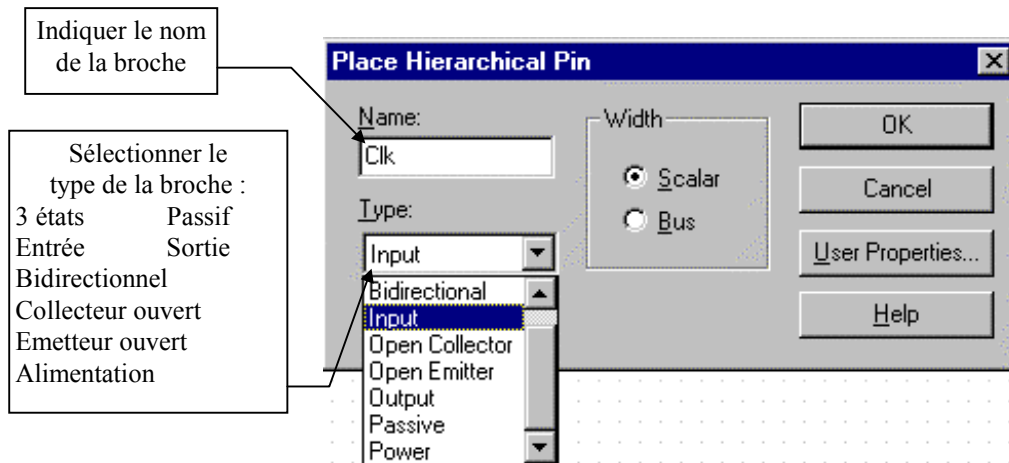
Le schéma principal (schéma racine) doit comporter au moins un bloc hiérarchique placé en choisissant l'option du menu Place - Hierarchical Block ou en cliquant sur l'icône  (Place hierarchical block).



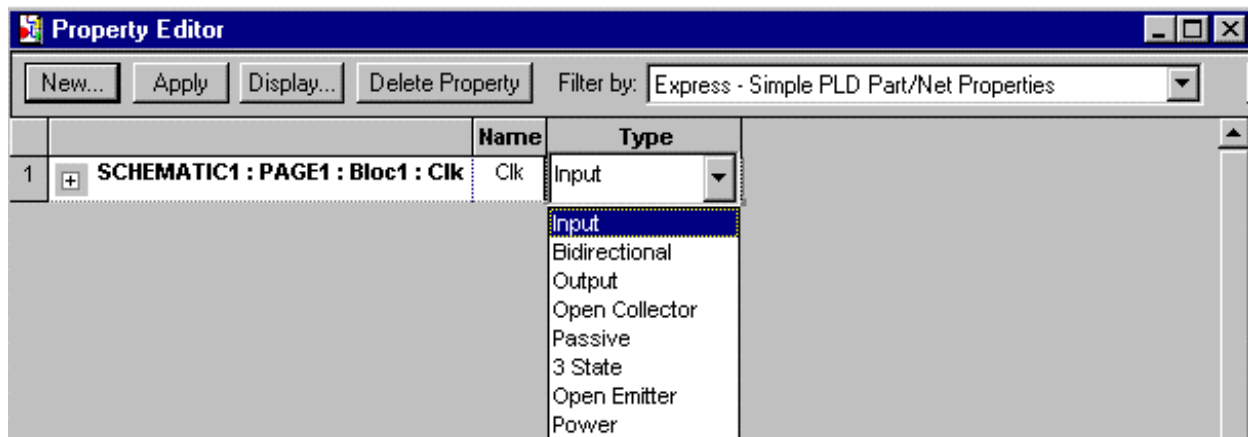
OrCad Capture attend maintenant que l'utilisateur dessine le contour du bloc hiérarchique.

🖍 Dessiner ensuite le bloc à la **bonne dimension** : celle-ci ne peut être modifiée par la suite !

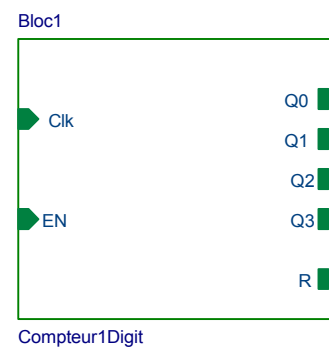
Sélectionner le bloc puis choisir l'option du menu Place - Hierarchical Pin ou cliquer sur l'icône  (Place hierarchical pin).



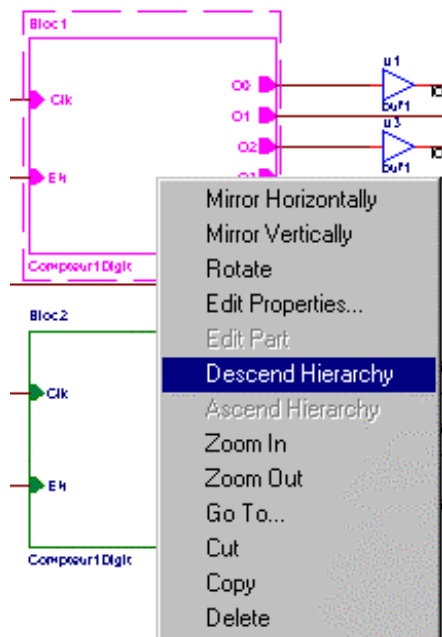
La modification du type est possible, après placement, en double cliquant sur la broche puis en utilisant l'éditeur de propriétés.



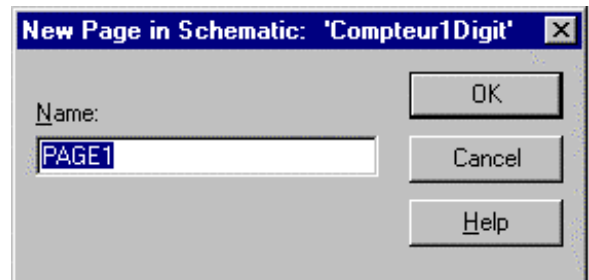
Le bloc hiérarchique doit avoir l'allure suivante :



Sélectionner le bloc, puis en cliquant sur le bouton droit de la souris, choisir l'option « Descend Hierarchy ».

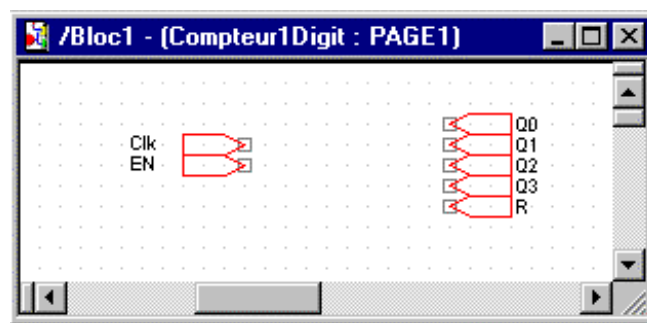


La fenêtre suivante demande un nom pour la page schéma (le nom « Page1 » par défaut convient ou peut éventuellement être remplacé par le nom de son choix).

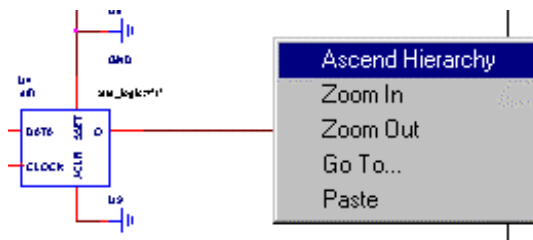


Le schéma comporte automatiquement les « Ports hiérarchiques » représentés sur le « Bloc hiérarchique ». Les « Ports » sont automatiquement :

- à gauche pour les entrées, alimentations, signaux passifs et bidirectionnels
- à droite pour les sorties, sorties 3 états et à collecteur ou émetteur ouvert (**malgré leur aspect**)

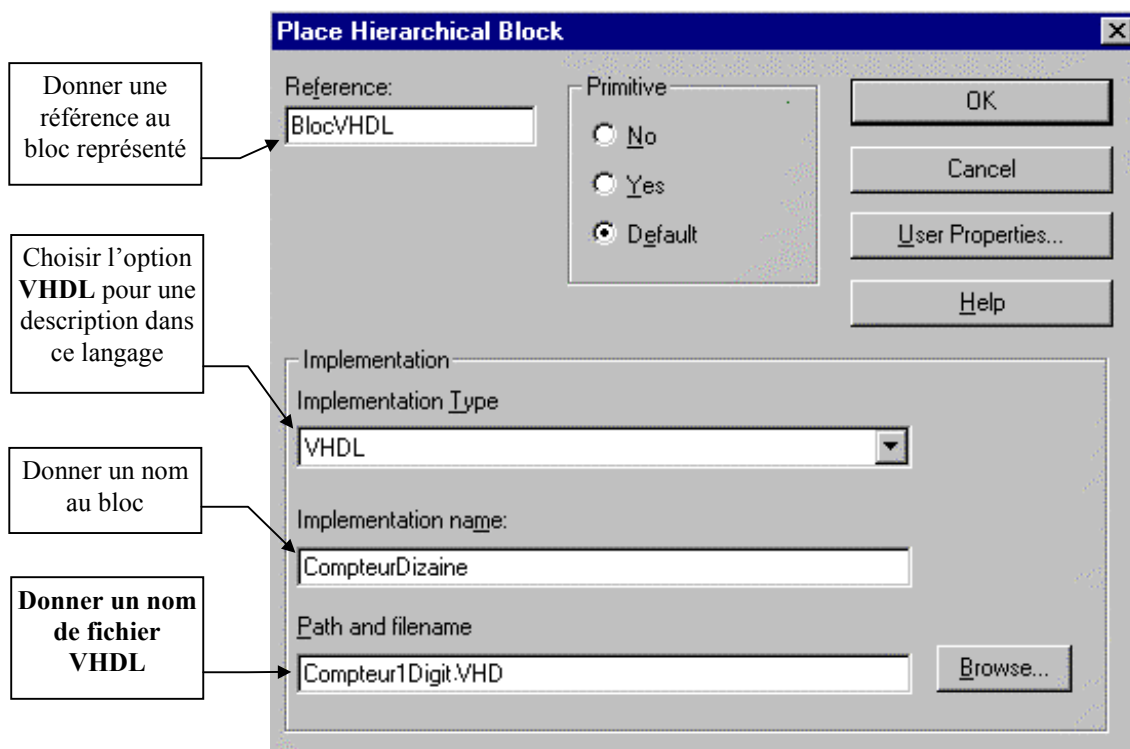


Le retour à la page schéma principale est possible en cliquant sur le bouton droit de la souris, puis en choisissant l'option « Ascend Hierarchy ».



1.4.2) Définition d'un bloc hiérarchique associé à une description en langage VHDL

La description de la fonction réalisée par un bloc peut être effectuée en langage VHDL.
Dans ce cas, la création du bloc hiérarchique diffère par les choix mentionnés ci-dessous :



Penser à spécifier l'extension du nom du fichier **.VHD**.

Définir ensuite les différentes broches hiérarchiques comme dans le paragraphe précédent.
En cliquant sur le bouton droit de la souris sur le bloc hiérarchique créé, choisir l'option « Descend Hierarchy ». L'ossature générale du fichier VHDL s'affiche avec les descriptions des entrées et des sorties définies dans le bloc hiérarchique.

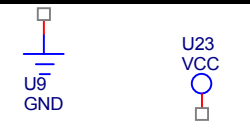
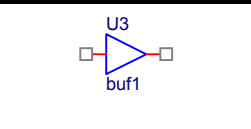
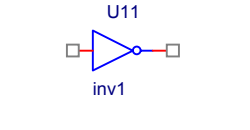
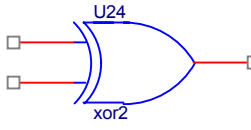
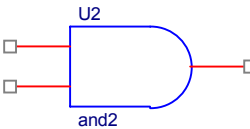
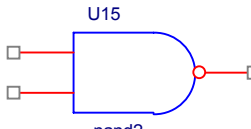
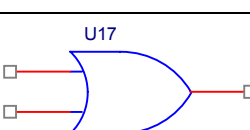
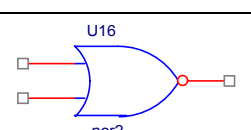
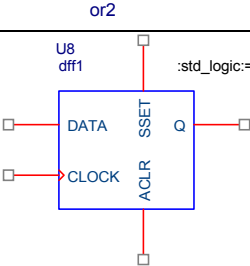
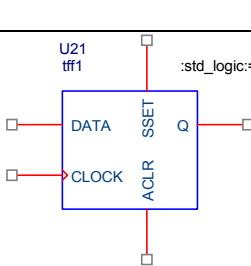
```
-- VHDL created by OrCAD Express
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.numeric_std; Use;

ENTITY CompteurDizaine is
  PORT (
    R      : OUT STD_LOGIC;
    Q0     : OUT STD_LOGIC;
    Q1     : OUT STD_LOGIC;
    Q2     : OUT STD_LOGIC;
    Q3     : OUT STD_LOGIC;
    Clk    : IN  STD_LOGIC;
    EN     : IN  STD_LOGIC);
END CompteurDizaine;

ARCHITECTURE behavior OF CompteurDizaine IS
BEGIN
  behavior;
END behavior;
```

1.5) Principaux symboles de base (Librairie SPLD)

Dans le cas d'un développement d'un «**Simple PLD**» (circuits PAL et GAL entre autres), la librairie de symboles utilisée s'appelle SPLD.OLB. Elle contient divers symboles. Parmi ceux-ci, les plus importants sont représentés dans le tableau suivant:

Symbole	Fonction	Variantes	Symbole	Fonction	Variantes
	Application d'un état logique 0 ou 1			Buffer	BUF2 BUF3 BUF4 BUF8 BUF12 BUF16 BUF24 BUF32
	Inverseur	INV2 INV3 INV4 INV8 INV12 INV16 INV24 INV32		Ou Exclusif	XOR3 XOR4 XOR8
	Et	AND3 AND4 AND8		Non Et	NAND3 NAND4 NAND8
	Ou	OR3 OR4 OR8		Non Ou	NOR3 NOR4 NOR8
	Bascule D	DFF2 DFF3 DFF4 DFF8 DFF12 DFF16 DFF24 DFF32		Bascule T	TFF2 TFF3 TFF4 TFF8 TFF12 TFF16 TFF24 TFF32


1.6) Contraintes relatives au brochage des circuits logiques programmables SPLD

Chaque broche externe d'un circuit logique programmable de la famille des SPLD doit comporter :

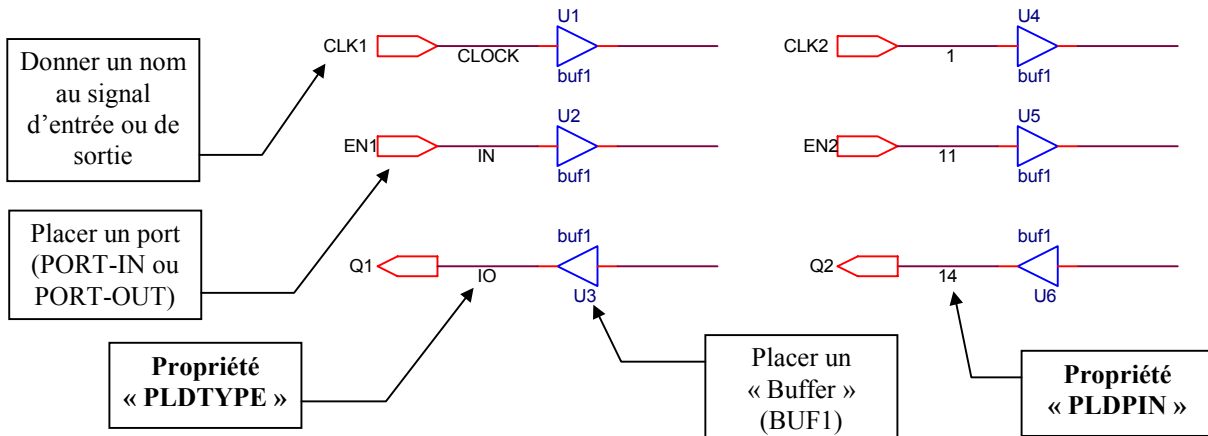
- un port : définit le sens de circulation de l'information (PORT-IN ou PORT-OUT) ;
- un nom pour chaque port : identifie pour le simulateur le nom du signal d'entrée ou de sortie ;
- un buffer : réalise l'interface entre l'extérieur du SPLD et la structure interne ;
- une liaison entre ces deux éléments (Port et Buffer) :

Cette liaison doit comporter une propriété définissant :

- soit le type de la broche :

 **IN / OUT / IO / CLOCK (attention à la syntaxe)**, dans le cas où le brochage du circuit peut être choisi automatiquement par le compilateur. La propriété à définir est « PLDTYPE » ;

- soit le numéro de la broche : 1 ... 24, dans le cas où le brochage du circuit est déjà imposé. La propriété à définir est « PLDPIN ».



Cet exemple mentionne les deux possibilités : PLDTYPE et PLDPIN. En général, une seule des propriétés est définie pour les broches de l'ensemble du projet.

Ces deux propriétés sont définies en suivant les étapes :

- sélectionner une liaison ou toutes les liaisons Port / Buffer , et seulement celles-ci !;
- cliquer avec le bouton droit de la souris (sur cette liaison) et choisir l'option « Edit Properties » ;
- sélectionner dans l'éditeur de propriétés le filtre « Express - Simple PLD Part/Net Properties » et éventuellement l'onglet « Schematic Nets » ;
- mentionner les propriétés pour répondre aux contraintes fixées (brochage imposé ou non, respect des types des broches en fonction du circuit) ;
- sélectionner l'ensemble de la colonne « PLDTYPE » ou « PLDPIN », cliquer sur le bouton « Display » puis choisir l'option « Value only » : ceci permet d'afficher sur le schéma la valeur de la propriété ;
- quitter l'éditeur de propriétés.

1) Editeur de propriétés

2) Sélectionner le filtre

3) Sélectionner l'onglet

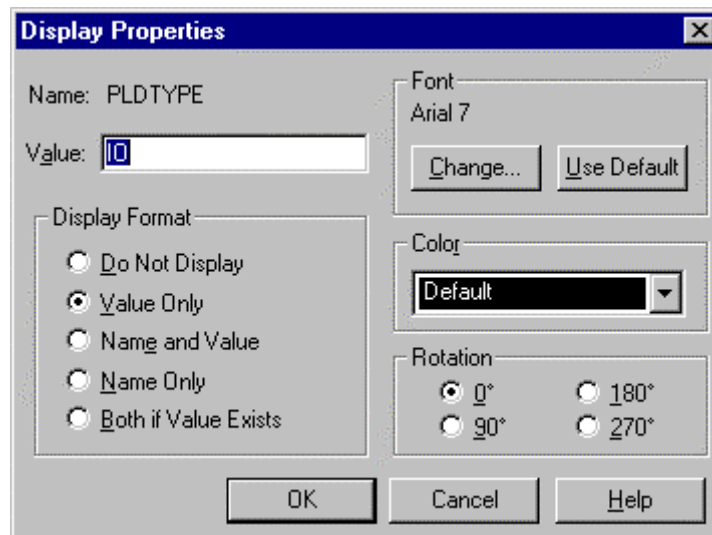
4) Spécifier les valeurs

5) Sélectionner la colonne

6) Cliquer sur Display

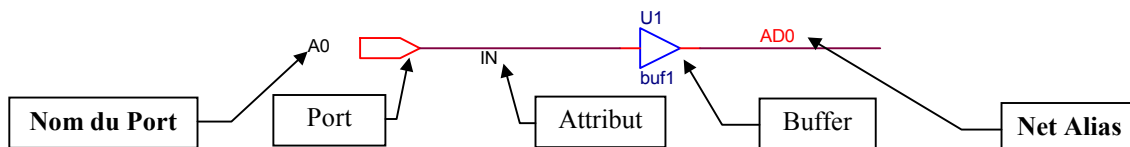
7) Choisir Value Only

La modification est ensuite possible directement sur le schéma en **double cliquant sur l'attribut** « PLDTYPE » ou « PLDPIN » :

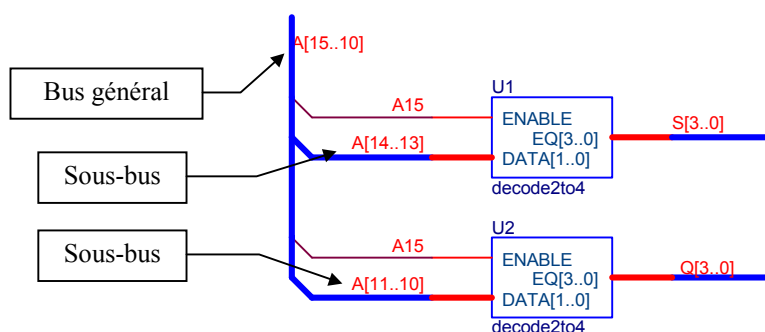



1.7) Contraintes relatives à la description de projet sous forme schématique

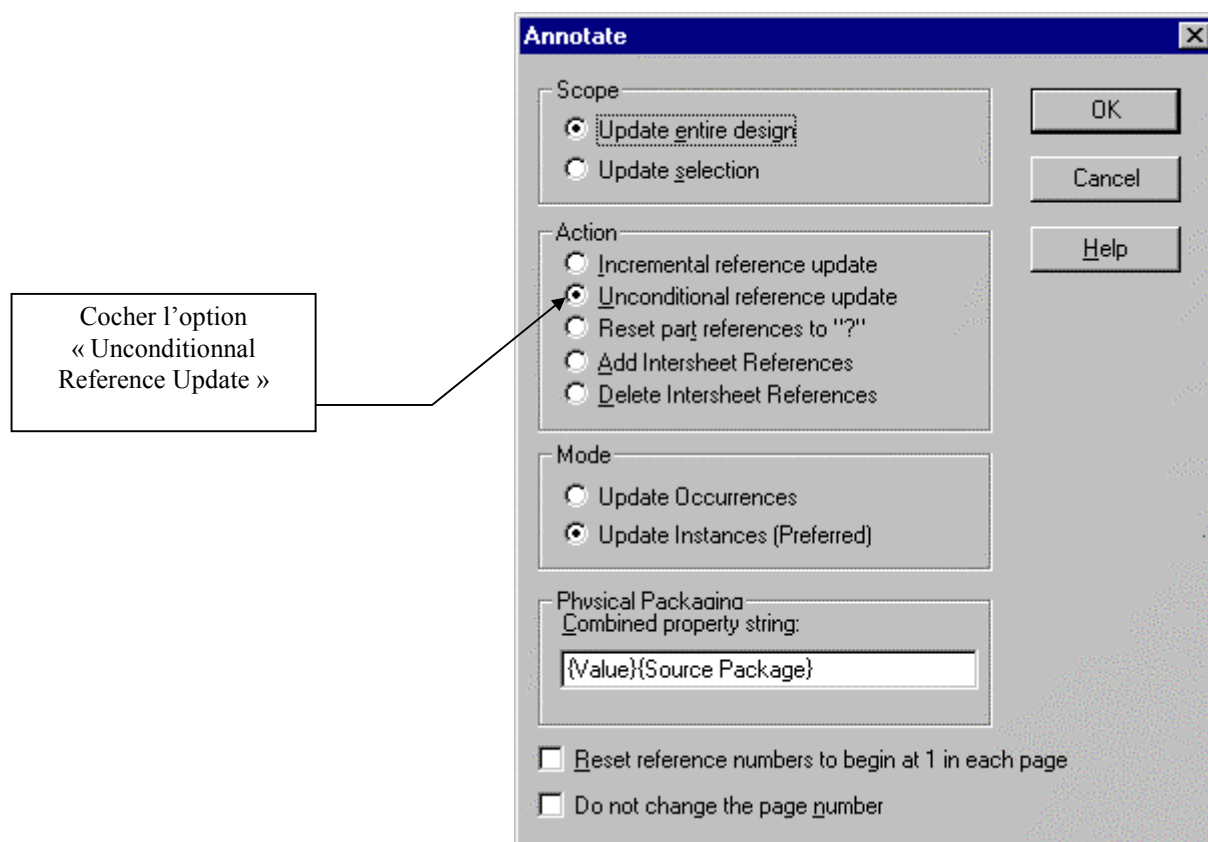
- Pour un circuit SPLD, seules les liaisons entre les Ports et les Buffers doivent comporter (obligatoirement) un attribut « PLDTYPE » ou « PLDPIN ».
- Pour un circuit SPLD, dans le cas où une liaison interne (autre que entre les Ports et les Buffers) comporte un attribut « PLDTYPE » ou « PLDPIN » (**MEME EFFACE**) une erreur de compilation sera générée par la suite.
- Pour un circuit SPLD, seules les entrées et sorties doivent comporter un Buffer.
- Il est éventuellement possible de définir un « Net Alias » sur la sortie d'un Buffer. Ce nom doit obligatoirement être différent du nom du Port d'entrée ou de sortie :



- Dans le cas de l'utilisation de bus, les règles suivantes doivent être respectées :
 - chaque bus doit comporter un « Net Alias » dont la syntaxe est **NomBus[i..j]** avec $i > j$ (sans espace)
 - chaque fil sortant du bus doit comporter un « Net Alias » dont la syntaxe est **NomBusN** avec $j \leq N \leq i$ (sans espace)
 - si plusieurs bus ont le même nom, ils doivent être reliés ensemble par un bus général



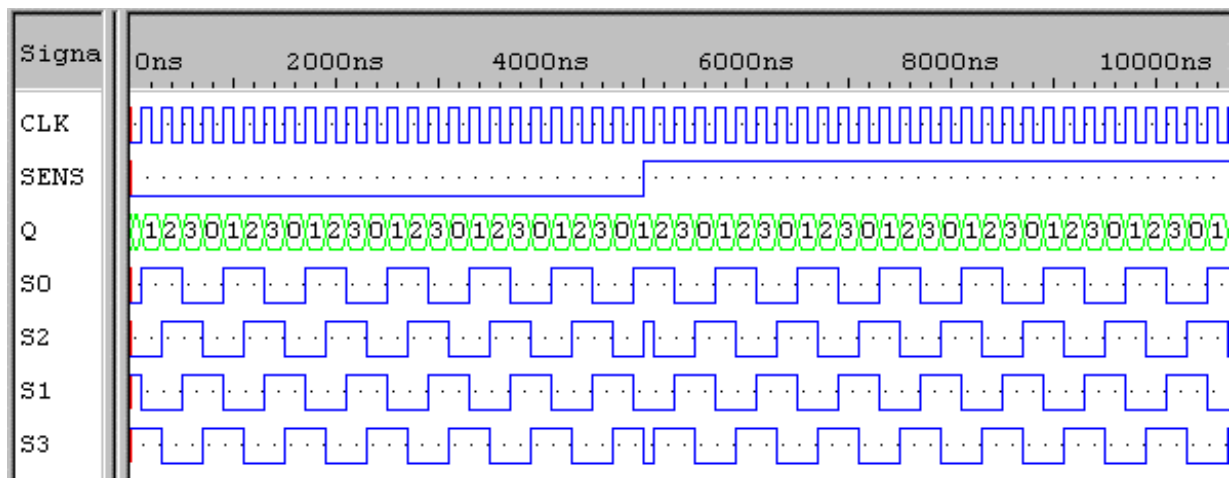
- Lors de la copie d'une portion de schéma (en faisant Edit Copy , puis Edit Paste), les références des composants ne sont pas modifiées : il y a donc plusieurs composants ayant le même repère. La renumérotation des composants doit obligatoirement se faire grâce à l'outil Annotate :
 - sélectionner dans le gestionnaire de projets le « Design »
 - choisir Tools - Annotate ou cliquer sur l'icône 



- Les symboles utilisés sur le schéma doivent OBLIGATOIREMENT provenir des bibliothèques spécifiques au type de circuit logique programmable utilisé (bibliothèque SPLD.OLB pour les SPLD par exemple). Ceci concerne aussi les symboles de masse (Gnd) et d'alimentation (Vcc).
- Après avoir lancé le simulateur OrCad Simulate, les états logiques aux différents points du schéma réalisé apparaissent directement sur celui-ci. Avant de l'imprimer, il est indispensable de **QUITTER OrCad Simulate**, de fermer la page schéma puis de la rouvrir.

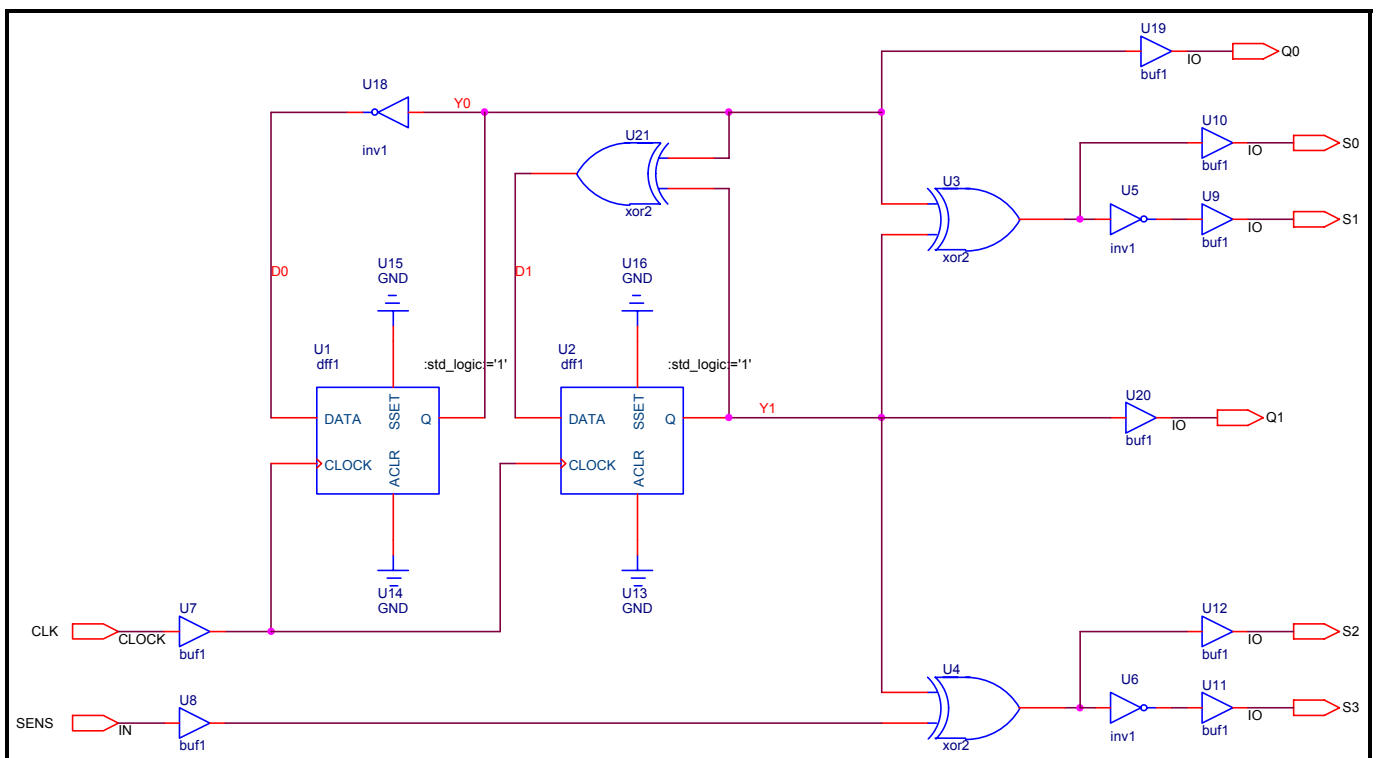
1.8) Exemples de projets destinés à des circuits logiques programmables SPLD

Les 4 exemples présentés ici sont 4 versions différentes d'un projet de commande de moteur pas à pas. Les chronogrammes souhaités, en fonction des entrées « Sens » et « Clk » sont :



1.8.1) Exemple de projet « Schéma simple »

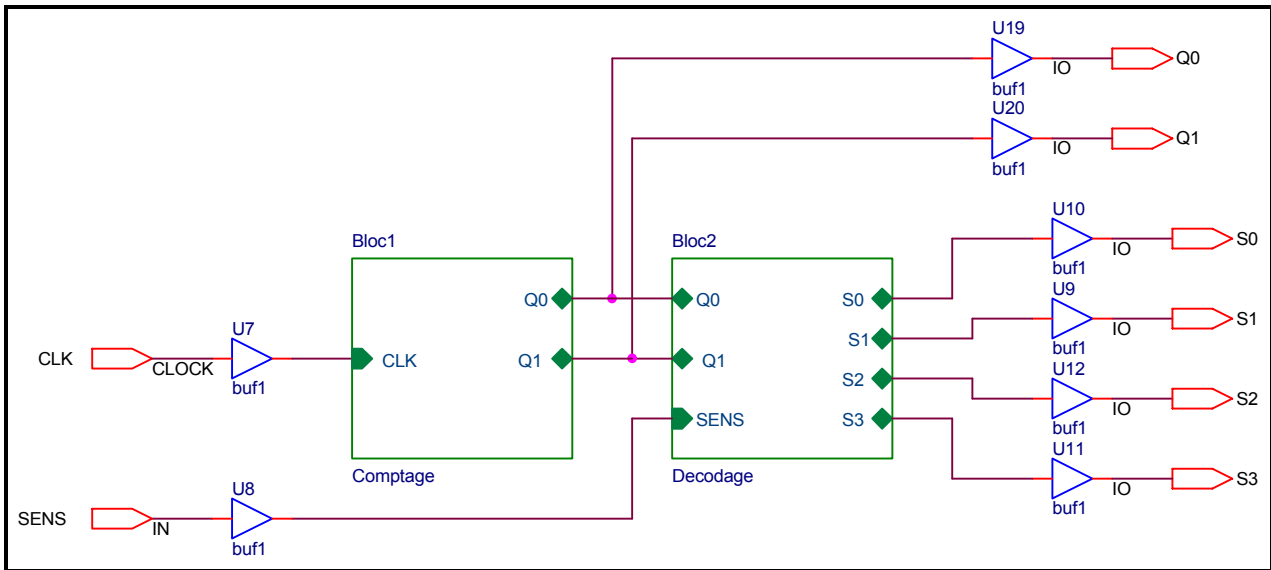
Un seul schéma représente l'ensemble du projet :



Ce type d'architecture convient pour de petits projets seulement. En effet, la lisibilité du schéma général devient laborieuse pour un projet plus conséquent.

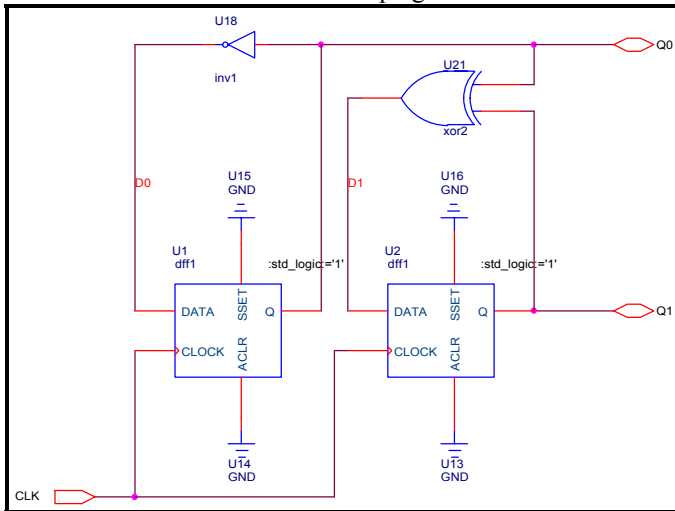
1.8.2) Exemple de projet « Schéma hiérarchique »

Un schéma « Racine » représente la partie brochage du circuit logique programmable.

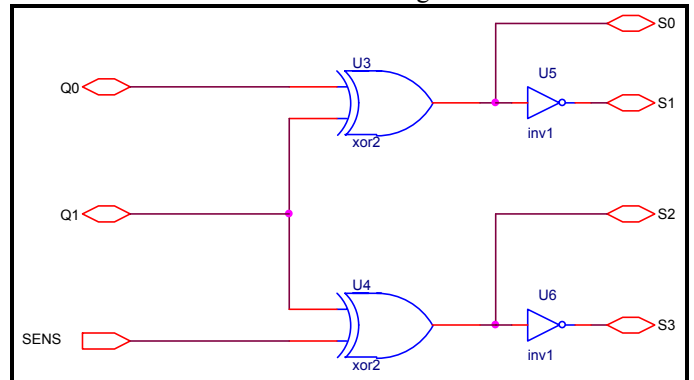


Les « blocs hiérarchiques » précédents correspondent à des schémas représentant les fonctions « Comptage » et « Décodage ».

Bloc1 : Comptage



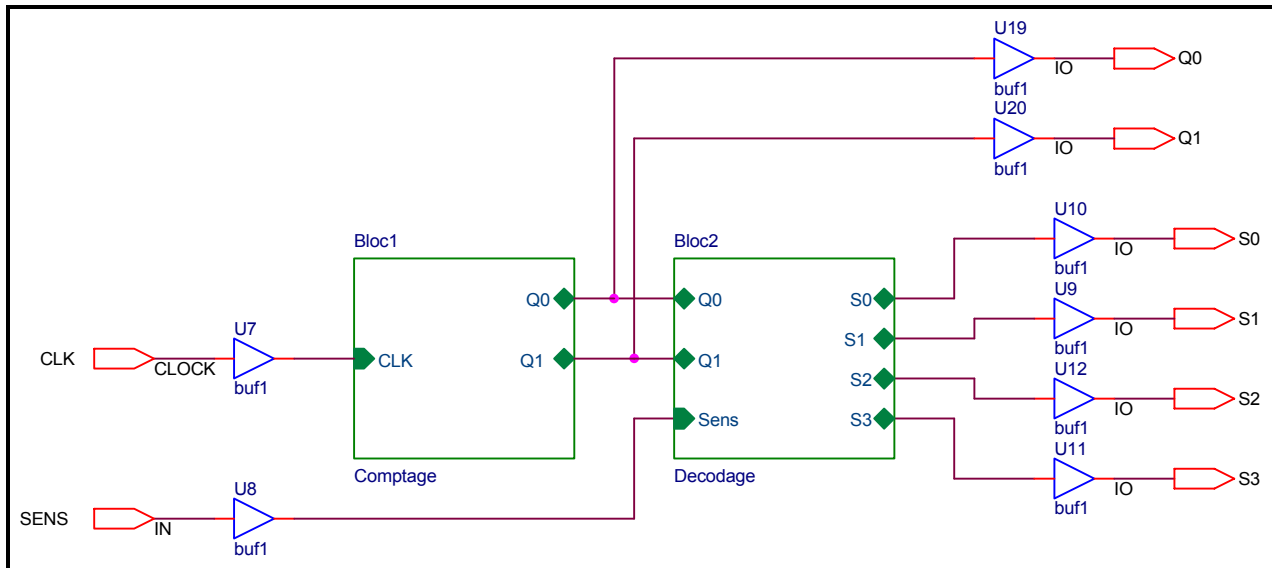
Bloc2 : Decodage



Ce type d'architecture convient pour des projets moyennement complexes, et plus particulièrement, dans le cas de fonctions combinatoires.

1.8.3) Exemple de projet « Schéma hiérarchique et code VHDL »

Un schéma « Racine » représente la partie brochage du circuit logique programmable.



Les « blocs hiérarchiques » précédents correspondent à des fichiers VHDL décrivant les fonctions « Comptage » et « Décodage ».

Bloc1 : Comptage	Bloc2 : Decodage
<pre>-- VHDL created by OrCAD Express Library ieee; Use ieee.std_logic_1164.all; Use ieee.numeric_std.all; -- librairie additionnelle pour l'incréméntation Use ieee.std_logic_arith.all; ENTITY Comptage is PORT (CLK : IN STD_LOGIC; Q0 : INOUT STD_LOGIC; Q1 : INOUT STD_LOGIC); END Comptage; ARCHITECTURE behavior OF Comptage IS -- Déclaration d'une variable auxiliaire pour le process signal Sortie : STD_LOGIC_VECTOR (1 DOWNTO 0); BEGIN process (Clk) begin if clk'event and clk = '1' -- détection d'un front montant then Sortie <= Sortie + 1; -- incréméntation de la variable end if; end process; Q0 <= Sortie (0); -- affectation des sorties Q1 <= Sortie (1); END behavior;</pre>	<pre>-- VHDL created by OrCAD Express Library ieee; Use ieee.std_logic_1164.all; Use ieee.numeric_std.all; ENTITY Decodage is PORT (S0 : INOUT STD_LOGIC; S1 : INOUT STD_LOGIC; S2 : INOUT STD_LOGIC; S3 : INOUT STD_LOGIC; Q0 : INOUT STD_LOGIC; Q1 : INOUT STD_LOGIC; Sens : IN STD_LOGIC); END Decodage; ARCHITECTURE behavior OF Decodage IS BEGIN S0 <= Q0 XOR Q1; S1 <= not (Q0 XOR Q1); S2 <= SENS XOR Q1; S3 <= not (SENS XOR Q1); END behavior;</pre>

Ce type d'architecture convient pour tous les types de projet. Les parties mentionnées en italique sont générées automatiquement par OrCad Express à partir du schéma. Le code VHDL à écrire (mentionné en gras) est relativement succinct.

1.8.4) Exemple de projet « Code VHDL seul »

Un fichier VHDL décrit l'ensemble des fonctions « Comptage » et « Décodage ».

```
-- VHDL created by OrCAD Express
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.numeric_std.all;
-- librairie additionnelle pour l'incrémentation
Use ieee.std_logic_arith.all;

ENTITY Compteur_Decodeur is
  PORT (
    CLK  : IN STD_LOGIC;
    Sens : IN STD_LOGIC;
    Q    : INOUT STD_LOGIC_VECTOR (1 DOWNTO 0);
    S0,S1,S2,S3 : INOUT STD_LOGIC);

-- définition du brochage du PAL 22V10
attribute pldtype : string;
attribute pldtype of Clk : signal is "CLOCK"; -- IN interdit !!
attribute pldtype of Sens : signal is "IN";
attribute pldtype of Q : signal is "IO";
attribute pldtype of S0,S1,S2,S3 : signal is "IO";

END Compteur_Decodeur;

ARCHITECTURE A_Compteur_Decodeur OF Compteur_Decodeur IS
-- Déclaration d'une variable auxiliaire pour le process
signal Sortie : STD_LOGIC_VECTOR (1 DOWNTO 0);

BEGIN
  process (Clk)      -- partie séquentielle : description de la fonction Comptage
  begin
    if clk'event and clk = '1' -- détection d'un front montant
    then Sortie <= Sortie + 1;   -- incrémentatation de la variable
    end if;
  end process;

  Q <= Sortie; -- affectation des sorties
  S0 <= Q(0) XOR Q(1);  --partie combinatoire : description de la fonction Décodage
  S1 <= not S0;
  S2 <= SENS XOR Q(1);
  S3 <= not S2;
END A_Compteur_Decodeur;
```

Ce type d'architecture convient pour tous les types de projet. Le code VHDL à écrire est constitué de différentes parties à reprendre à partir d'exemples.

2) Compilation du projet

Une fois le projet terminé, schémas et/ou descriptions en langage VHDL complétés, il faut traduire celui-ci en fichier JEDEC en vue de programmer le circuit logique. Ceci nécessite 2 ou 3 étapes : Vérification syntaxique du langage VHDL (§2.1), Compilation (§2.2) et Routage (§2.3).

2.1) Vérification syntaxique (du langage VHDL)

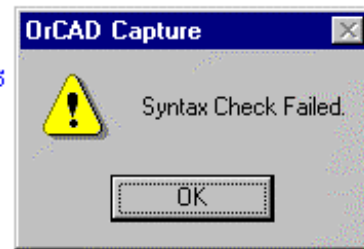
Dans le cas d'une description en langage VHDL, une première vérification de syntaxe s'impose.

Choisir l'option du menu Edit - Check VHDL Syntax ou raccourci clavier « Alt + C ».

Ceci permet de détecter des erreurs de **syntaxe** du langage VHDL. Le texte est sélectionné sur ou après l'erreur détectée.

```
END CompteurDizaine;
```

```
ARCHITECTURE behavior OF CompteurDizaine IS
BEGIN
  R <= Q3 and not Q2 and not Q1 and Q0
END behavior;
```




Dans cet exemple, il manque un point virgule à la fin de la ligne d'affectation.

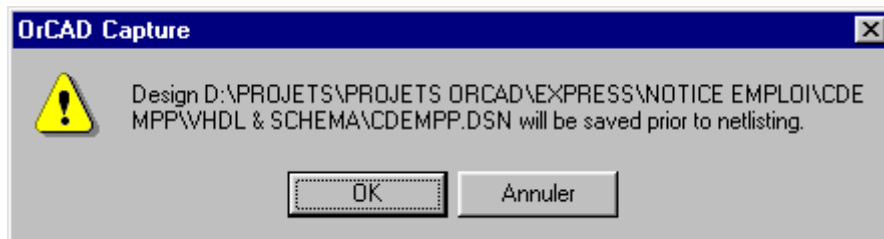
2.2) Compilation du projet

La compilation consiste à vérifier la faisabilité du projet effectué **pour une famille générique** de circuits logiques programmables, **sans tenir compte de la référence exacte du circuit**.

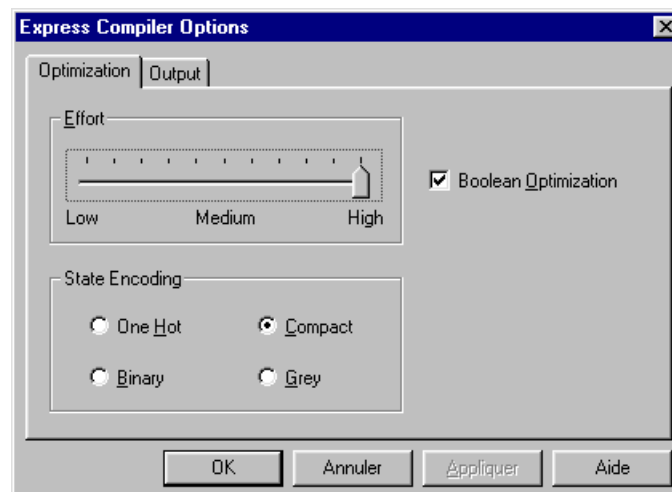
La compilation nécessite auparavant l'enregistrement des fichiers du projet.

La compilation est exécutée en sélectionnant le **gestionnaire de projets** puis en choisissant l'option du menu Tools - Compile... ou en cliquant sur l'icône  (Compile).

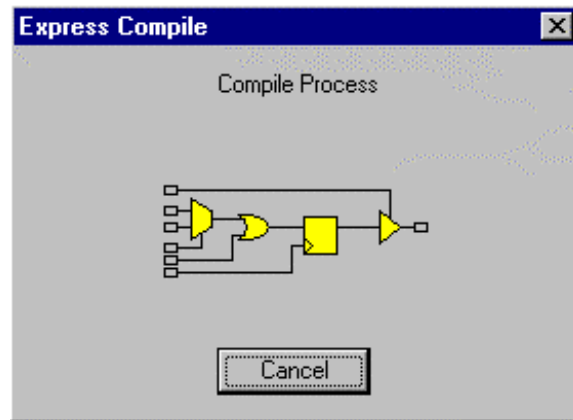
Un message demande confirmation en cas d'oubli d'enregistrement :



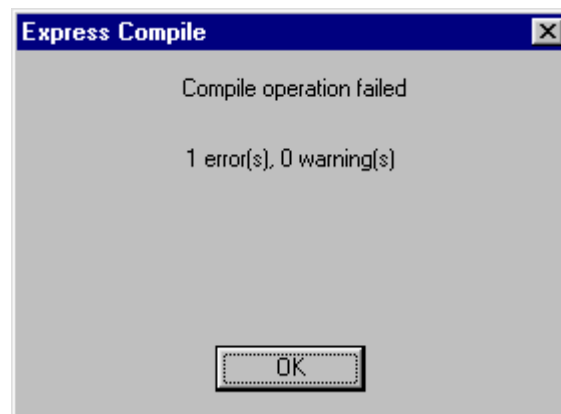
Lorsque la compilation est lancée à partir du menu TOOLS, la fenêtre suivante apparaît. Cliquer OK.



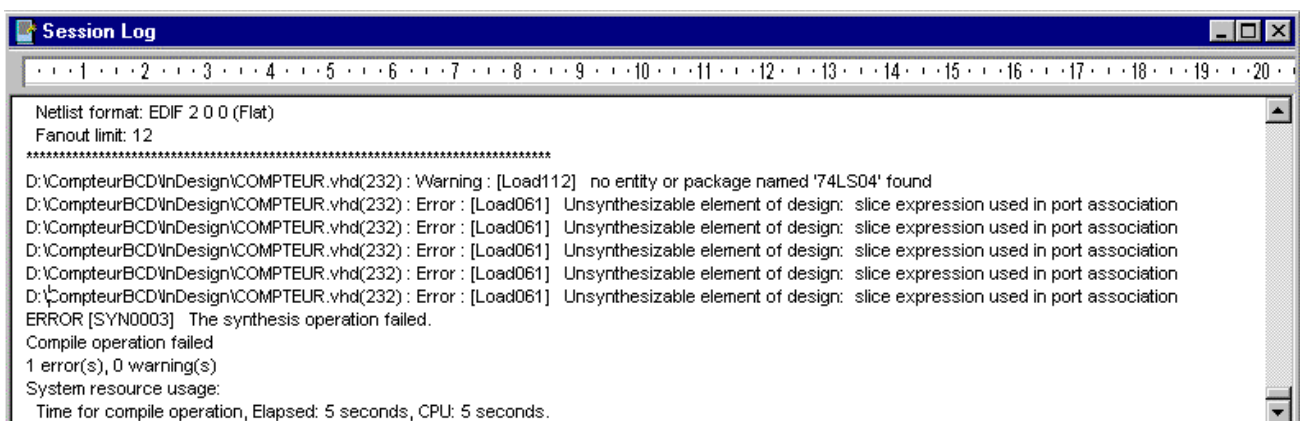
Le processus de compilation est ensuite exécuté :



Dans le cas où des erreurs de compilation sont détectées, un message signale la présence d'erreurs ou d'avertissements.



Dans ce cas, il est nécessaire d'examiner la nature des erreurs dans la fenêtre de « Session Log ». Si celle-ci est accessible à l'écran, l'activer pour la faire passer au premier plan. Si elle n'est pas visible, choisir l'option du menu Window - Session Log...




Dans cet exemple, un inverseur de technologie TTL (74LS04) a été utilisé à la place d'un inverseur issu de la librairie SPLD.OLB.

Remarque : il est possible, à ce stade, d'effectuer une simulation dite « In Design ». Celle-ci est uniquement fonctionnelle. Se reporter au § 3.1 pour la simulation.

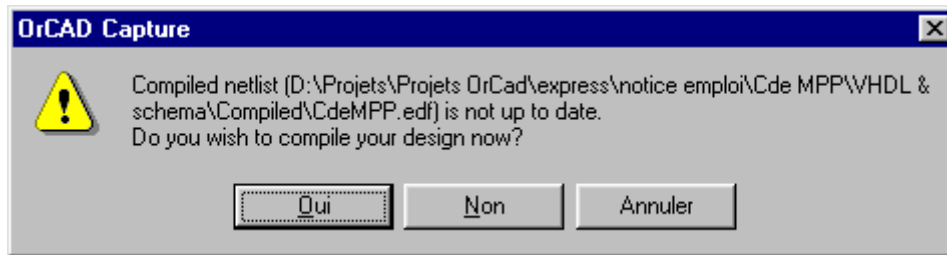
2.3) « Routage » du circuit

Le « routage » du circuit utilise un outil logiciel (« Fitter ») spécifique au fabricant du circuit logique programmable employé. Cela consiste à déterminer les liaisons à effectuer à l'intérieur du circuit en tenant compte de ses ressources internes (nombre et nature des opérateurs de base, liaisons possibles, nature des broches d'entrées/sorties ...).

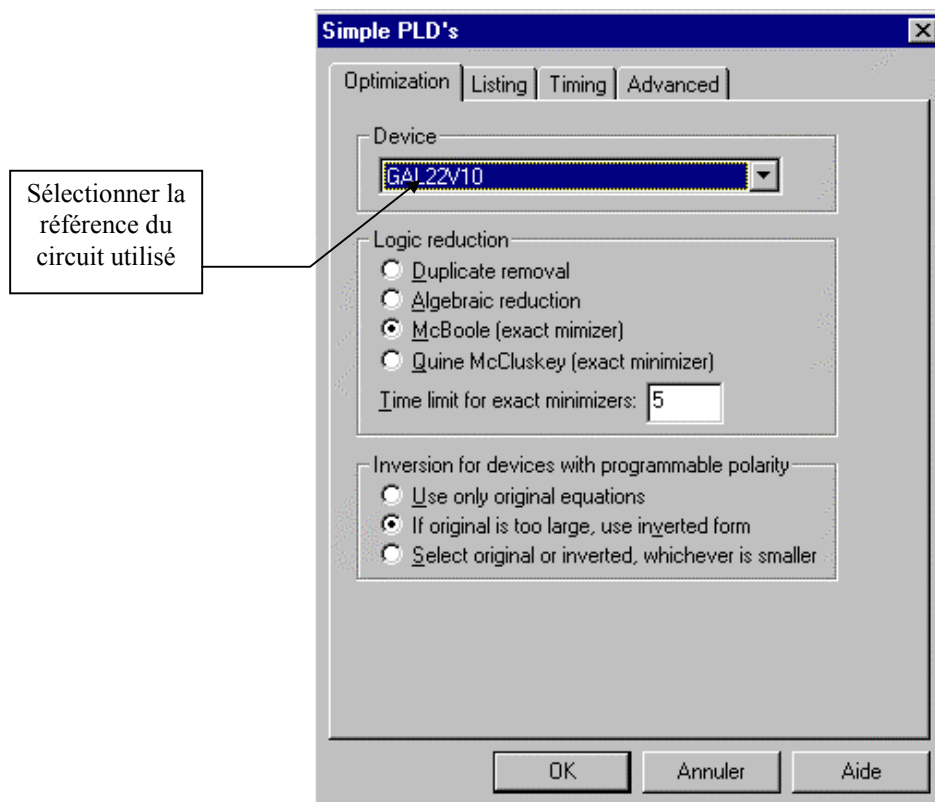
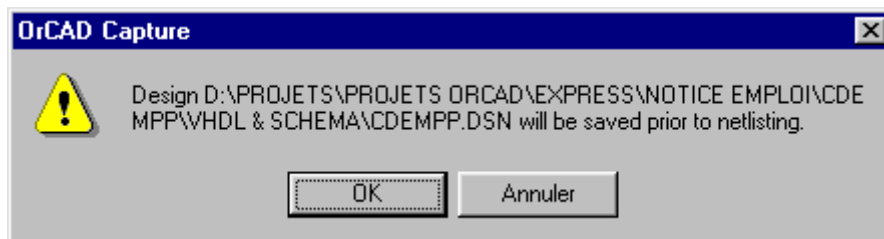
Le « routage » nécessite auparavant l'enregistrement des fichiers du projet ainsi que la compilation.

Le routage est exécuté en sélectionnant le **gestionnaire de projets** puis en choisissant l'option du menu Tools - Build... ou en cliquant sur l'icône  (Build).

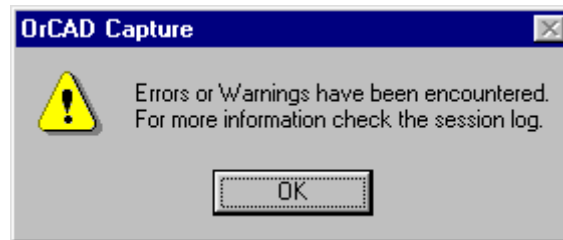
Un message demande confirmation en cas d'oubli de compilation :



Un message demande confirmation en cas d'oubli d'enregistrement :



Dans le cas où des erreurs de « routage » sont détectées, un message signale la présence d'erreurs ou d'avertissement.



Dans ce cas, il est nécessaire d'examiner la nature des erreurs dans la fenêtre de « Session Log ». Si celle-ci est accessible à l'écran, l'activer pour la faire passer au premier plan. Si elle n'est pas visible, choisir l'option du menu Window - Session Log...

```

*****
OrCAD Simple PLD Fitter tools
*****
OrCAD PLD FITTER x1.11 11/5/98 (Source file D:\CompteurBCD\Timed\Compteur.jed)
Copyright (C) 1988-1998 OrCAD All Rights Reserved.
1. Parsing input file
2. Loading device tables
3. Converting pin connections
   Group 4 (CLOCK)
   Group 1 (IN)
   Group 3 (IO)
4. Expanding source statements
   Signal N00041
   Signal EXP_N246
   Signal Q0
.....
5. Reducing equations
   Signal EXP_N239
   Signal Q0
NOTE (PLD0230) Signal EXP_N353 shares a macrocell with signal Q0.
NOTE (PLD0230) Signal EXP_N349 shares a macrocell with signal Q1.
.....
6. Configuring architectural fuses
7. Generating fuse array
ERROR (PLD2766) Signal Q1 does not have a valid clock portion.
8. Checking for error conditions
NOTE (PLD0200) One fatal errors found in input file.
NOTE (PLD0201) No warnings.

```

← Erreur signalée

Dans cet exemple, la réalisation d'un compteur asynchrone avec un GAL22V10 est impossible. L'interprétation des messages d'erreurs (« Error ») ou d'avertissements (« Warning ») n'est pas toujours évidente !

3) Simulation


La simulation permet de vérifier, avant programmation effective du circuit, que le projet réalisé répond effectivement au cahier des charges fixé.

3.1) Lancement du simulateur

La simulation nécessite auparavant :

- l'enregistrement des fichiers du projet ;
- la compilation du projet, pour une simulation fonctionnelle (« In Design ») ;
- le routage du circuit, pour une simulation temporelle (« Timed »).

Elle est exécutée en sélectionnant le **gestionnaire de projets** puis en choisissant l'option du menu Tools -

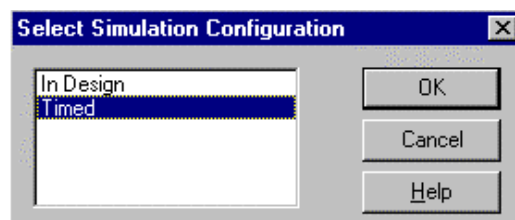
Simulate... ou en cliquant sur l'icône  (Simulate).

Eventuellement, des messages de confirmation sont affichés, en cas :

- d'oubli de compilation.
- d'oubli d'enregistrement.
- d'oubli de routage.

Une fenêtre demande le type de simulation désirée :

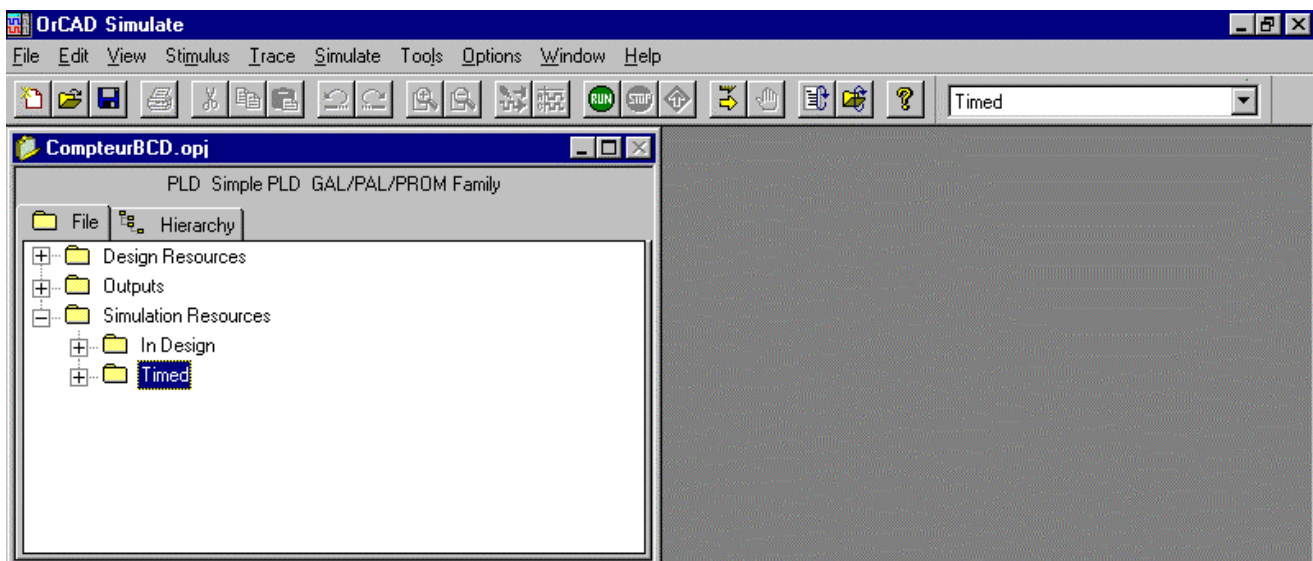
- In Design : effectue une simulation fonctionnelle du circuit (sans tenir compte du brochage retenu, des temps de propagation...).
- Timed : tient compte des caractéristiques du circuit (interconnexions réalisées, temps de propagation...).



Le simulateur se lance automatiquement.

Sa fenêtre est constituée de :

- une barre de menu
- une barre d'icônes
- une copie du gestionnaire de projets (*presque* identique à celui de OrCad Capture).




Remarque : Dans certains cas, après avoir déjà effectué une simulation, un message demande confirmation du chargement du projet, puis un second pour les stimulus. Répondre « OK » dans les 2 cas.

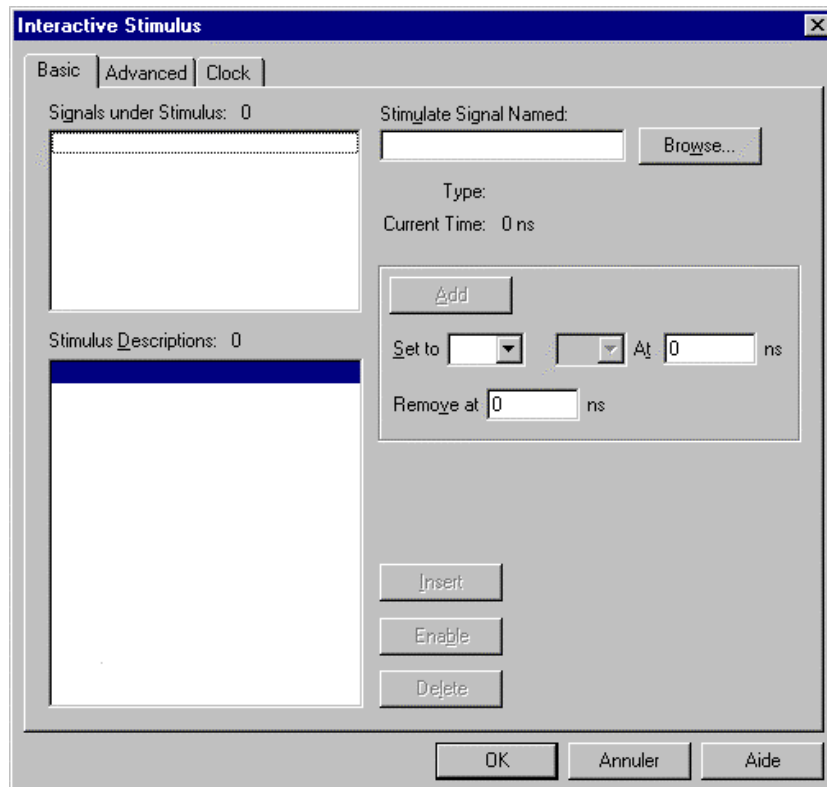
3.2) Définition des signaux d'entrée

Pour effectuer une simulation, il est nécessaire de définir les chronogrammes des signaux d'entrée (« Stimulus ») grâce à l'éditeur de Stimulus.

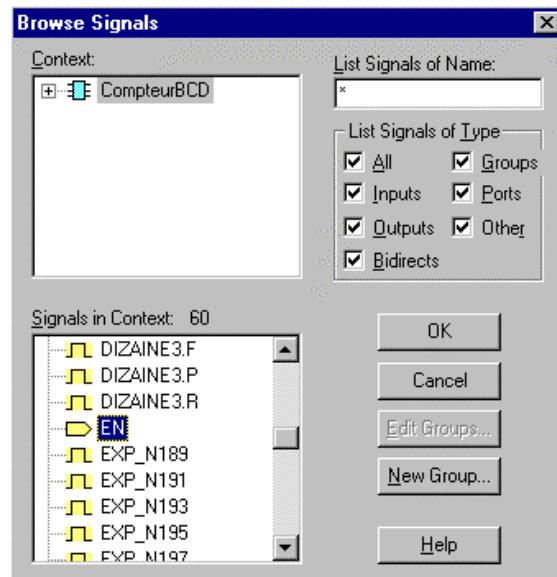
3.2.1) Editeur de stimulus

L'éditeur de stimulus se lance :

- lors de la **première création** de stimulus : en choisissant l'option du menu Stimulus - **New Interactive**...
- lors de la **modification de stimulus** : en choisissant l'option du menu Stimulus - **Edit Interactive**...ou en cliquant sur l'icône  (Edit Stimulus).




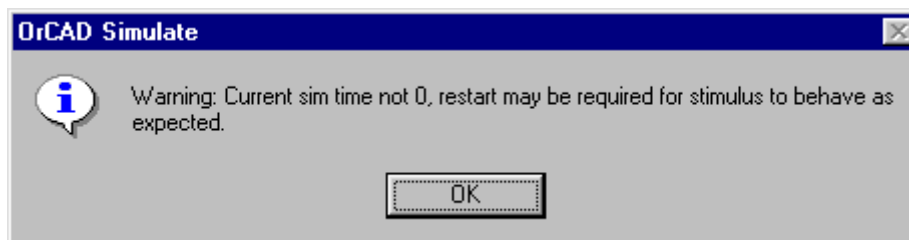
- Choisir l'onglet « Basic » ou « Advanced » ou « Clock » suivant la nature du stimulus à définir.
 - « Basic » : stimulus simple (spécification temporelle des passages à 0 / 1 /...) voir exemple 1 ;
 - « Clock » : stimulus de type horloge voir exemple 2 ;
 - « Advanced » : stimulus complexe (gestion de bus, incrémentation, décrémentation, boucles diverses...) voir exemple 3.
- Cliquer sur le bouton « Browse » pour choisir le signal à définir.



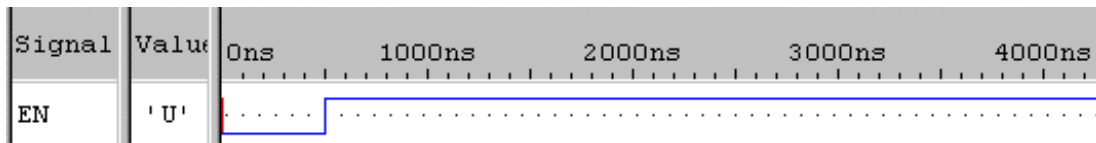
- Définir éventuellement un groupe de signaux correspondant à un bus (voir § 3.3.3)
- Sélectionner le nom du signal à définir puis valider par « OK »
- Définir les caractéristiques du stimulus.

Remarque : Après avoir effectué des modifications sur les stimulus, un message signale l'obligation de réinitialiser le simulateur avant d'effectuer une nouvelle simulation. Ceci s'effectue en choisissant l'option

du menu **Simulate - Restart** ou en cliquant sur l'icône  (Restart).



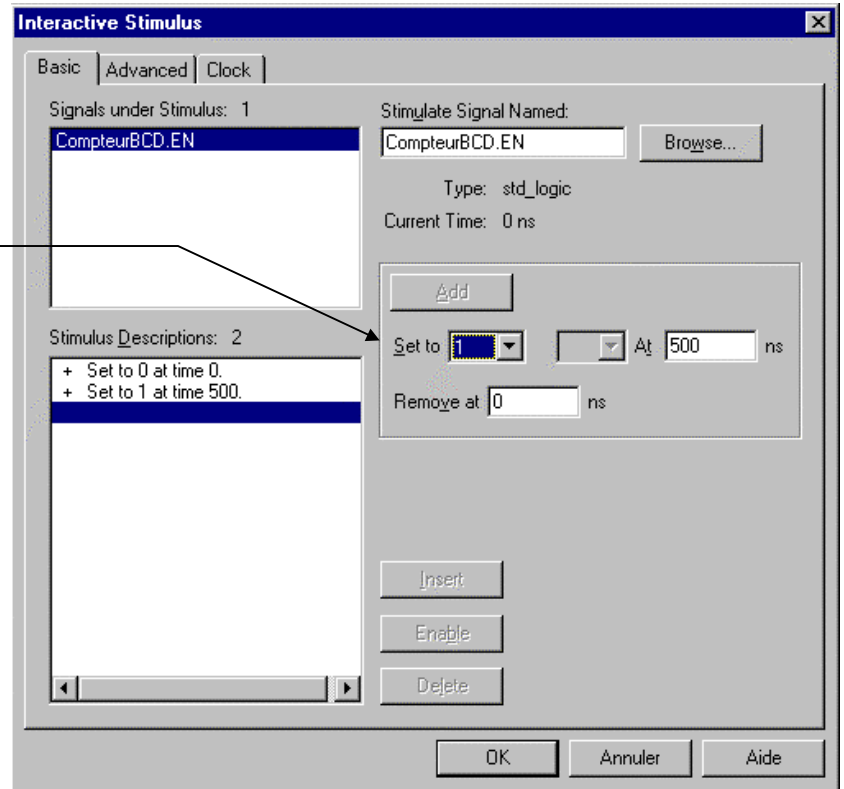
Exemple 1 : Signal de validation (de type « Basic ») inactif pendant les 500 premières ns.



mise à 0 à t = 0 Set to 0 At 0 ns ADD
 mise à 1 à t=500ns Set to 1 At 500 ns ADD

Il est ensuite possible :

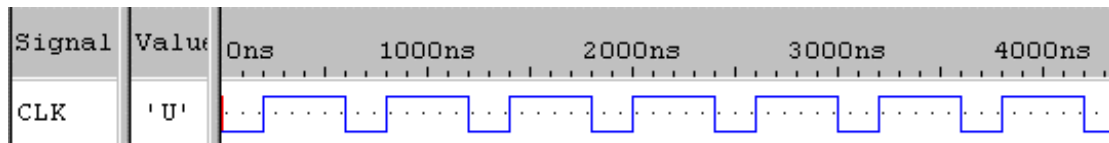
- de modifier une ligne du stimulus : **sélectionner** la ligne, mentionner les nouvelles valeurs, puis cliquer sur « Change »
- d'ajouter une ligne en cliquant sur le bouton « Insert »
- de supprimer une ligne : sélectionner la ligne inutile, puis cliquer sur le bouton « Delete »



Pour définir un autre stimulus de même type, cliquer sur « Browse » ou choisir un autre onglet pour un autre stimulus de type différent.

Pour terminer l'édition des stimulus, cliquer sur « **OK** », puis sauvegarder les stimulus (voir §3.2.3).

Exemple 2 : Signal d'horloge (de type « Clock ») de rapport cyclique 66,6%.



définition t = 0 Start at **0** ns

mise à 0 Set to **0** For **200** ns

mise à 1 Set to **1** For **400** ns

répétition infinie Repeat forever

validation **ADD** (1^{ère} fois) ou **CHANGE** (modification)

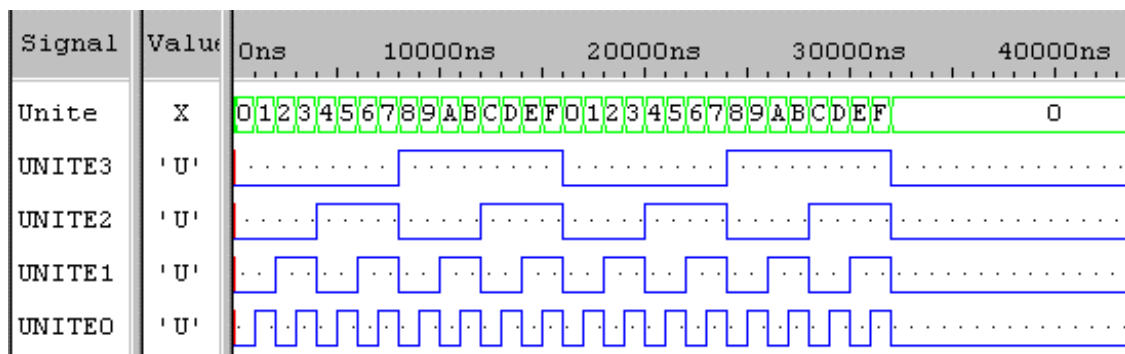
Il est ensuite possible :

- de modifier une ligne du stimulus : **sélectionner** la ligne, mentionner les nouvelles valeurs, puis cliquer sur « Change »
- d'ajouter une ligne en cliquant sur le bouton « Insert »
- de supprimer une ligne : sélectionner la ligne inutile, puis cliquer sur le bouton « Delete »

Pour définir un autre stimulus de même type, cliquer sur « Browse » ou choisir un autre onglet pour un autre stimulus de type différent.

Pour terminer l'édition des stimulus, cliquer sur « **OK** », puis sauvegarder les stimulus (voir §3.2.3).

Exemple 3 : Signal complexe (de type « Advanced ») incrémentation automatique d'un bus de 4 bits.



Définition à t = 0 :
Set to **0** **SET**

définition d'une boucle :
Repeat Block **32** times **Repeat Block**

durée entre 2 états :
Wait for **1000** ns **Wait for**

définition de l'incrément :
Increment by **1** Dec **Increment**

Il est ensuite possible :

- de modifier une ligne du stimulus : **sélectionner** la ligne, mentionner les nouvelles valeurs, puis cliquer sur le bouton correspondant
- d'ajouter une ligne en cliquant sur le bouton « Insert »
- de supprimer une ligne : sélectionner la ligne inutile, puis cliquer sur le bouton « Delete »

Interactive Stimulus

Basic Advanced Clock

Signals under Stimulus: 1
CompteurBCD.Unite

Stimulate Signal Named: CompteurBCD.Unite Browse...

Type: std_logic_vector
Current Time: 0 ns
Start at: 0 ns

Repeat Block: 32 times

Set to: 0 Dec

Wait for: 1000 ns

Increment: by 1 Dec

Decrement: by Dec

Remove

Insert Disable All

Delete

OK Annuler Aide

Attention : Dans certains cas la valeur « Start at » n'est pas égale à 0 par défaut. Il est alors nécessaire de la modifier.

Pour définir un autre stimulus de même type, cliquer sur « Browse » ou choisir un autre onglet pour un autre stimulus de type différent.

Pour terminer l'édition des stimulus, cliquer sur « OK », puis sauvegarder les stimulus (voir §3.2.3).

3.2.2) Groupes de signaux - Bus

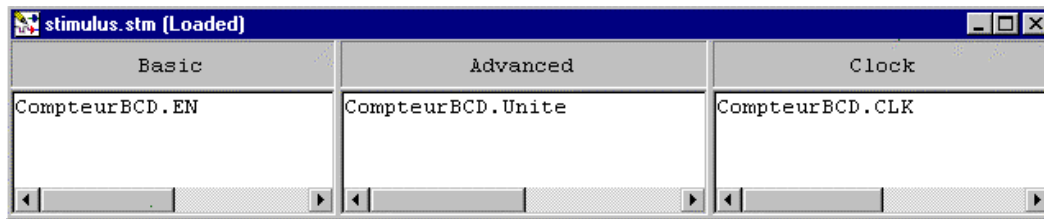
Certains signaux peuvent être regroupés en bus (appelés « Groups » dans OrCad Simulate). Voir §3.3.3) pour définir des groupes.

3.2.3) Sauvegarde des stimulus

Une fois les stimulus définis, il est nécessaire de sauvegarder le fichier les contenant.

Sélectionner la fenêtre des stimulus (ci-dessous) puis choisir l'option du menu File - Save ou cliquer sur

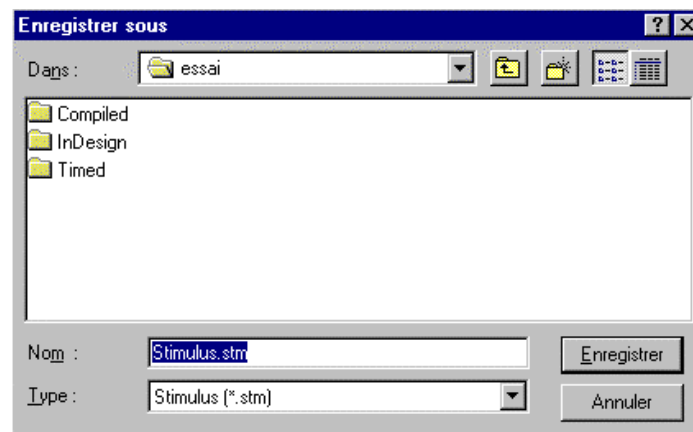
l'icône  (Save).



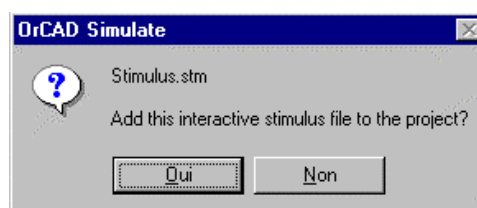
Une fenêtre demande le nom du fichier pour les stimulus. Le nom par défaut (Stimulus.stm) convient dans le cas où un seul type de simulation est utilisé.

Attention : le répertoire par défaut dans lequel sera enregistré le fichier schéma n'est pas systématiquement le répertoire du projet. Il est indispensable de parcourir l'arborescence pour retrouver le répertoire contenant le projet.

Dans le cas où le nom du fichier de stimulus est modifié, préciser l'extension .STM




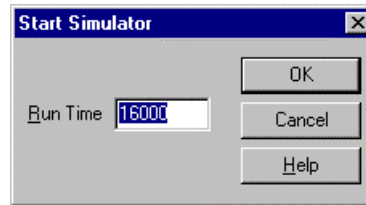
La fenêtre suivante demande s'il est nécessaire d'ajouter le fichier de stimulus au projet : répondre « Oui ».



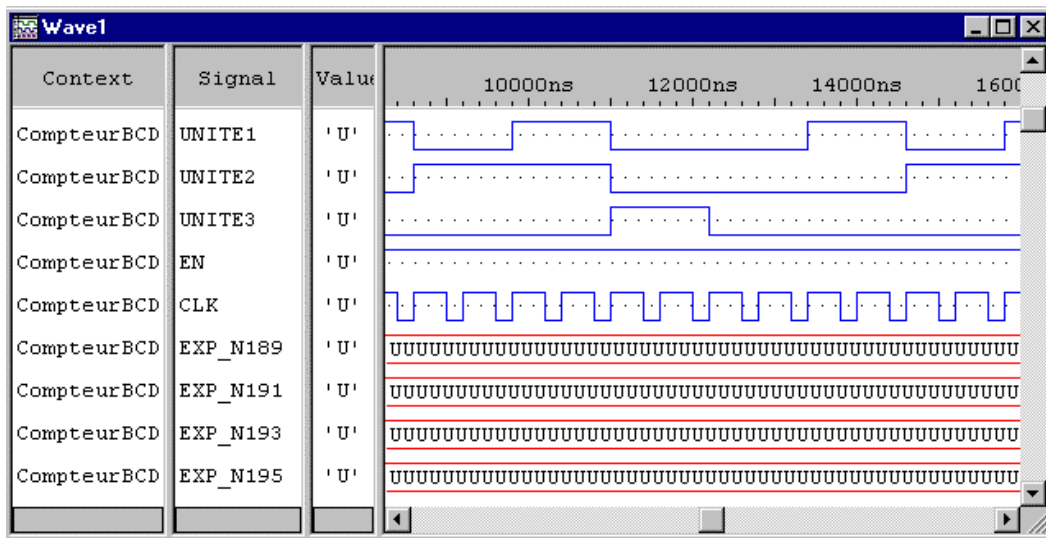
3.3) Simulation du projet

3.3.1) Lancement - Durée de simulation

La simulation est lancée en choisissant l'option du menu **Simulate - Run...** ou en cliquant sur l'icône  (Run). Dans le premier cas, une fenêtre demande la durée de la simulation (exprimée en ns).



Une fenêtre intitulée « Wave1 » apparaît : elle représente tous les signaux (externes et internes du circuit).



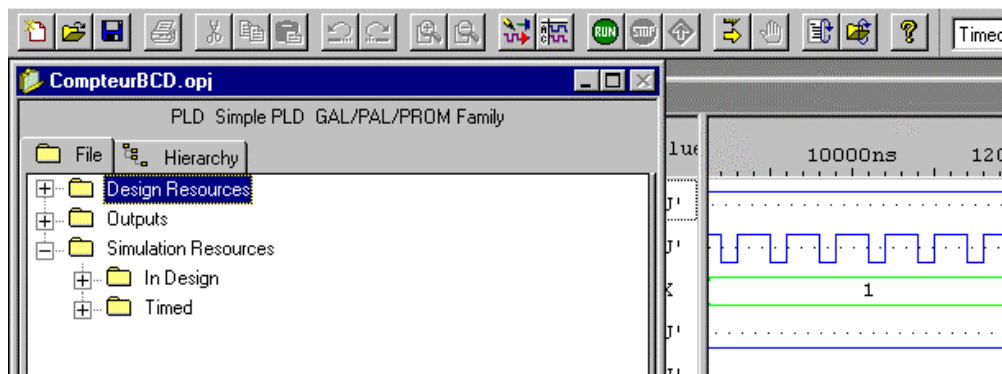
3.3.2) Choix des signaux à visualiser

La sélection des signaux désirés (Entrées et Sorties en général) s'effectue en sélectionnant dans l'une des colonnes (« Context », « Signal » ou « Value ») les différents signaux inutiles puis en les supprimant par la touche « Suppr » du clavier.

La sauvegarde de la liste des signaux utiles à visualiser nécessite :

- la sélection, dans le gestionnaire de projets, du dossier « Design Resources »
- puis la sauvegarde du projet en choisissant l'option du menu **File - Save** ou en cliquant sur

l'icône  (Save).

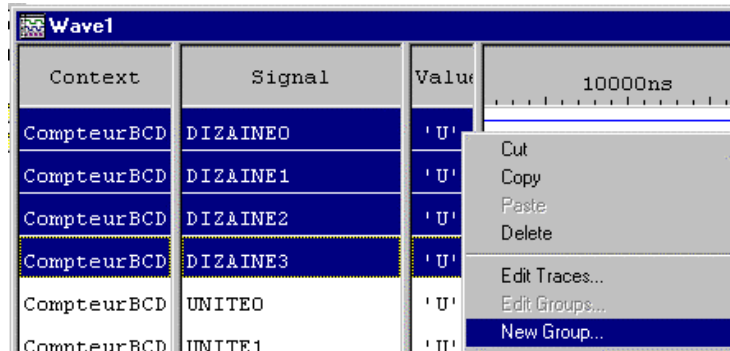


3.3.3) Groupes de signaux - Bus

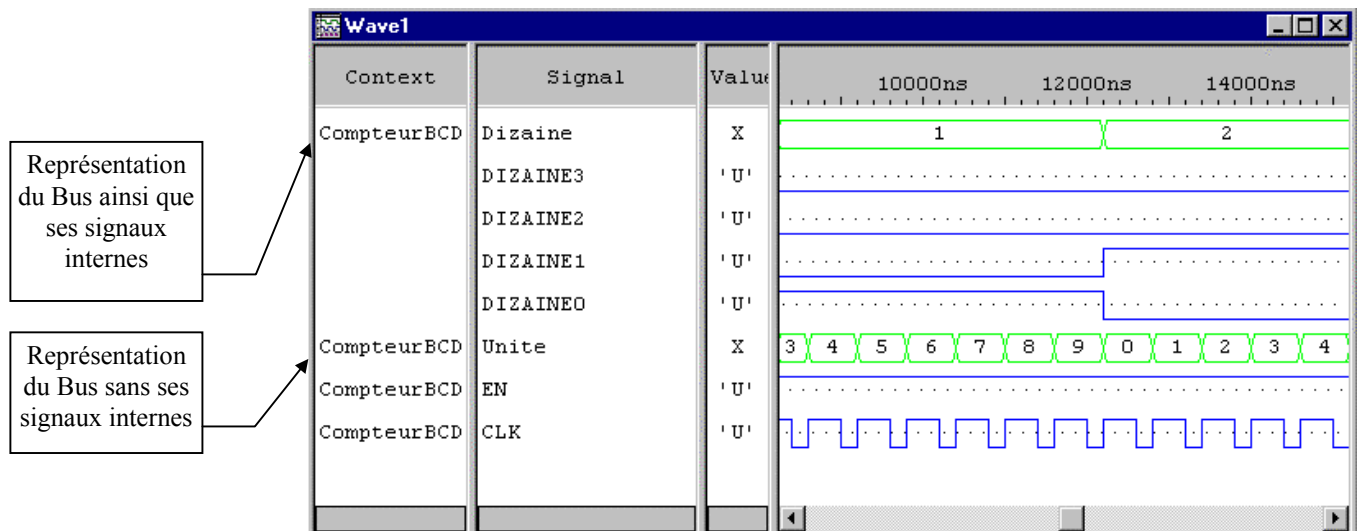
La représentation de certains signaux peut parfois être plus simple en les regroupant en bus (appelés « Group » dans OrCad Simulate).

Définition des groupes à partir de la fenêtre des résultats de simulation (« Wave1 »)

- sélectionner dans l'une des colonnes (« Context », « Signal » ou « Value ») les différents signaux à regrouper, en utilisant la touche CTRL en même temps que le clic de souris sur chaque signal;
- cliquer le bouton droit de la souris et choisir l'option « New Group... » pour créer un nouveau groupe ou l'option « Edit Group... » pour modifier un groupe existant ;
- donner un nom au groupe de signaux.

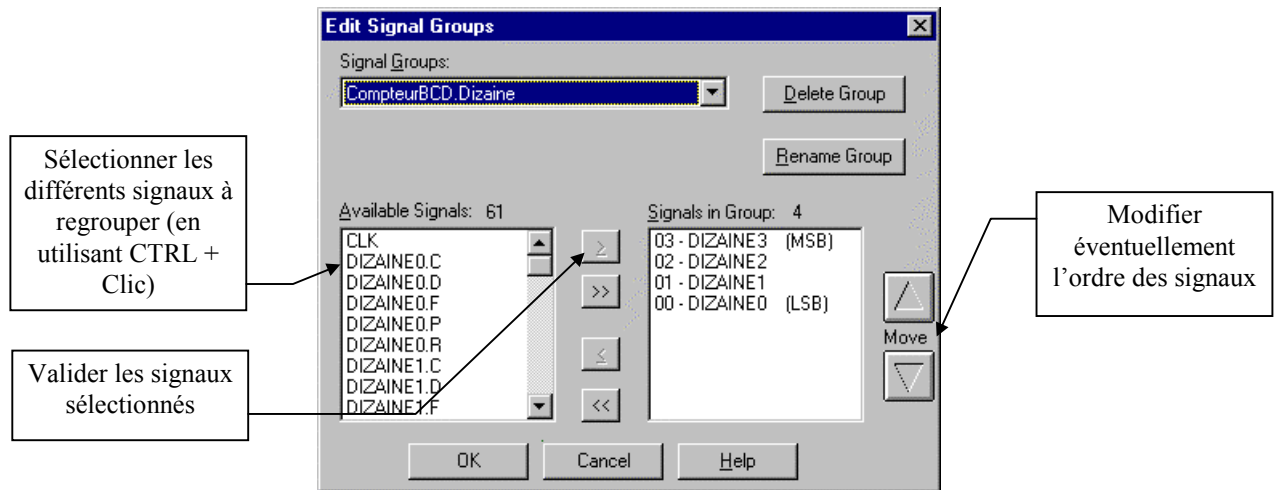


Les signaux sont maintenant représentés en un seul chronogramme que l'on peut développer ou non par un double clic sur le chronogramme.



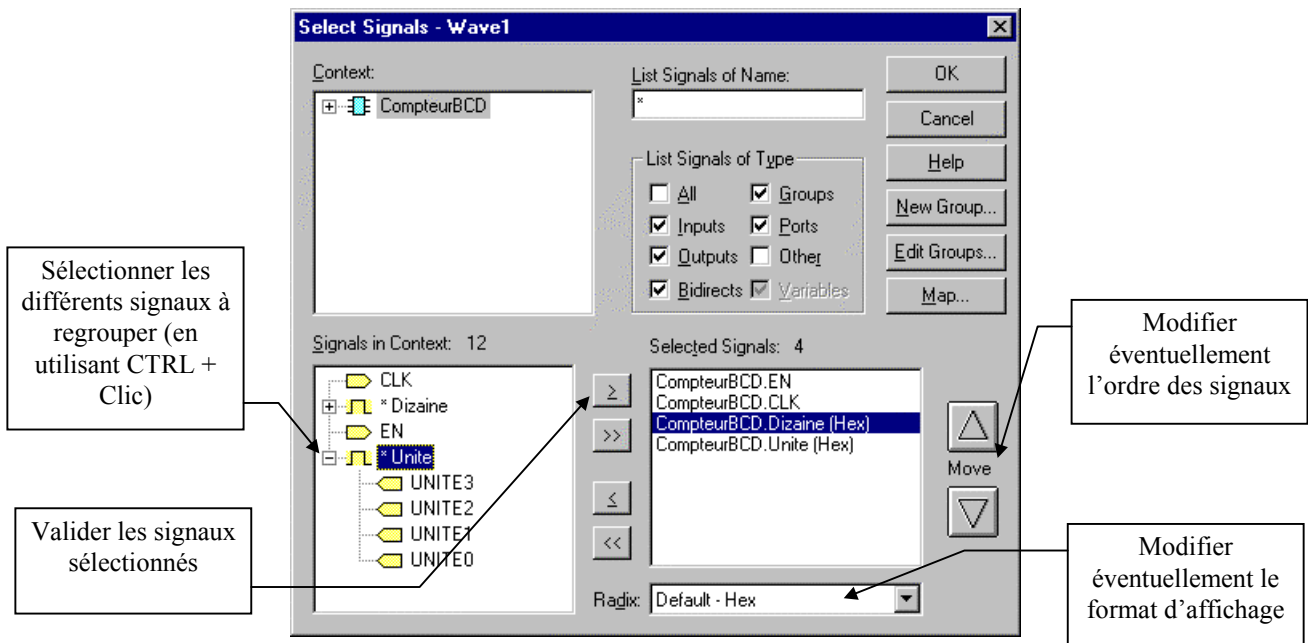
Définition des groupes à partir du menu « Stimulus »

Choisir les options : Stimulus - Edit interactive - Browse - Edit Groups (pour une modification) ou New Group (pour une création) pour les signaux d'entrée : voir page suivante

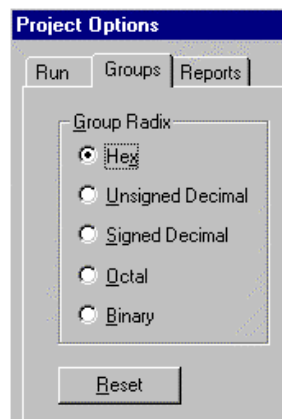


Définition des groupes à partir du menu « Trace »

Choisir les options : Trace - Edit Signal Trace - Browse - Edit Groups (pour une modification) ou New Group (pour une création) pour les signaux à visualiser.



Remarque : le format d'affichage par défaut des bus peut être modifié en choisissant l'option du menu Options - Project puis en sélectionnant l'onglet « Groups ».



3.3.4) Mesures de durées - Curseurs

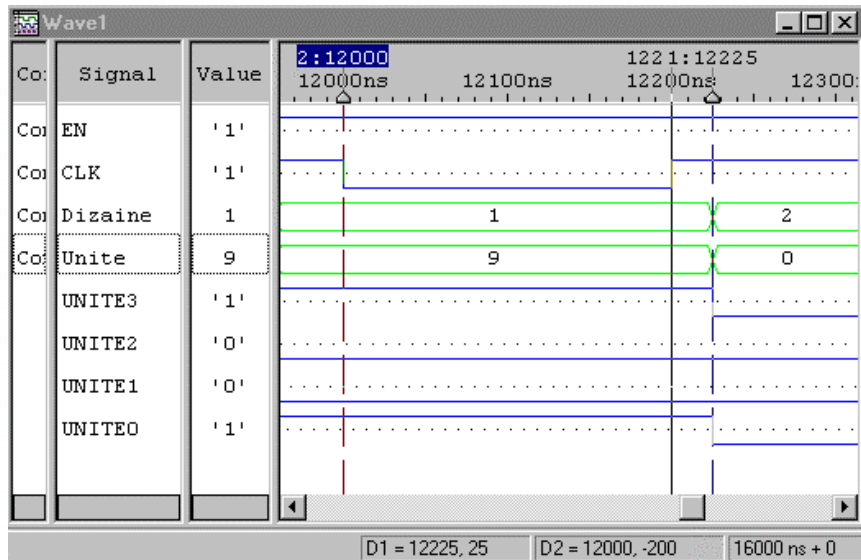
En cliquant dans la fenêtre de chronogramme un curseur vertical s'affiche.

Il est possible d'ajouter 2 autres curseurs (appelés « Delta Marker ») en cliquant sur le bouton droit de la souris et choisissant l'option « Add Delta Marker ».

La date pointée par le « Delta Marker » et l'écart par rapport au curseur principal s'affiche dans le bas de la fenêtre.

Le déplacement des « Delta Marker » s'effectue en déplaçant le mini triangle dans la barre des durées

Attention : le déplacement du mini triangle en dehors de la barre des durées supprime le « Delta Marker ».



3.4) Modification du projet


Dans le cas où une erreur de fonctionnement est constatée, une modification doit être apportée sur le projet (dans un schéma et/ou dans un fichier VHDL).

Il n'est pas indispensable de quitter le simulateur : réactiver dans la barre des tâches OrCad Capture.

Effectuer toutes les modifications nécessaires et relancer le processus de compilation :

- éventuellement, vérification syntaxique pour le langage VHDL
- compilation du projet
- routage du circuit.

De retour dans OrCad Simulate, il est nécessaire de recharger le nouveau projet en choisissant l'option du menu

Simulate - Reload Project ou en cliquant sur l'icône  (Reload).

4) Quitter le simulateur

Si les différents fichiers ont été correctement sauvegardés (Stimulus §3.2.3, contenu des différents fichiers du gestionnaire de projets §3.3.2, liste des courbes affichées §3.3.2), il n'y a aucune demande de confirmation en quittant OrCad Simulate.

Dans le cas contraire, répondre affirmativement aux différentes demandes de sauvegarde.

5) Impressions : Schémas - Code VHDL - Résultats de simulation

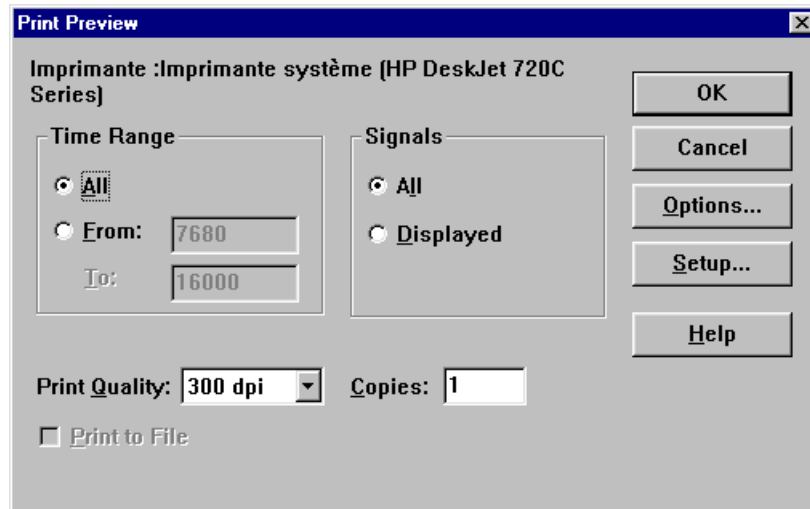
L'impression d'un schéma, d'un fichier VHDL ou des résultats de simulation nécessite obligatoirement l'utilisation de l'option du menu File - Print Preview.

En effet, la configuration de l'impression n'est pas la même suivant la nature de ce qui doit être imprimé :

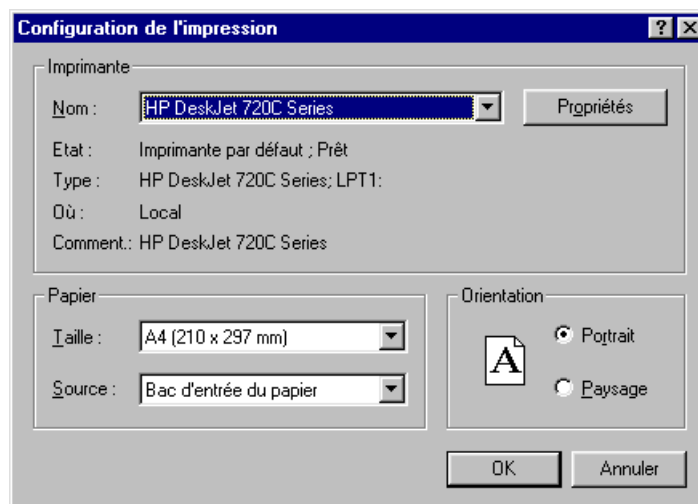
- schéma : orientation paysage ;
- fichier VHDL : orientation portrait ;
- chronogrammes de simulation : orientation paysage.

De plus, dans tous les cas il est possible de modifier les options d'impression (chronogrammes forcés sur une seule page, schéma représenté à l'échelle ou non...).

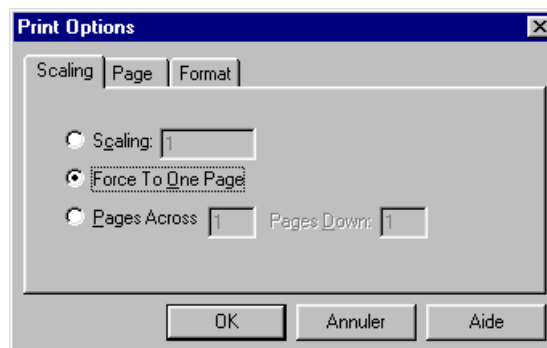
Le choix de l'option File - Print Preview affiche la fenêtre suivante :



Le bouton « Setup... » permet de choisir l'imprimante et de configurer le format (vertical / horizontal).



Le bouton « Options... » permet de forcer l'impression des chronogrammes sur une seule page.



Remarque : Après avoir lancé le simulateur OrCad Simulate, les états logiques aux différents points du schéma réalisé apparaissent directement sur celui-ci. Avant de l'imprimer, il est indispensable de **QUITTER OrCad Simulate**, de fermer la page schéma puis de la rouvrir.