

# Syst de Télécom Embarqués

## Chapitre IV: Simulation des modèles VHDL



- Simulation des circuits logiques à l'aide de ModelSim
- Utilisation du "Graphical Waveform Editor" du simulateur ModelSim
- Démarche à suivre
- Visualisation des sorties correspondant aux entrées

# Simulation VHDL

Désign du projet: Soit la porte And dont le code VHDL est:

```
entity andcircuit is
    port (    x, y : IN Bit;
           z : OUT Bit );
end entity;
architecture andcircuit_arch of andcircuit is
begin
    z  $\leftarrow$  x and y;
end architecture;
```

Créer le projet andcircuit.vhd

# Simulation VHDL

Désign du projet: Soit le code VHDL pour le test du circuit and:

```
entity andcircuit_test is
end entity;
architecture andcircuit_test_arch of andcircuit_test is
component andcircuit is
port ( x, y : IN Bit;
      z : OUT Bit );
end component;
signal testX,testY,testZ: bit;
begin
inst: andcircuit port map (testX,testY,testZ);
process
begin
testX <= '1';
testY <= '1';
wait for 50 ns;
```

# Simulation VHDL

```
testX <= '1';  
testY <= '0';  
wait for 50 ns;  
testX <= '0';  
testY <= '1';  
wait for 50 ns;  
testX <= '0';  
testY <= '0';  
wait for 50 ns;  
end process;  
end architecture;  
Cr  er le projet andcircuit_test.vhd
```

# Simulation VHDL

- Créez le dossier GRT dans la partition D
- Mettez les deux projets dans GRT
- File → New → Project (fig1)

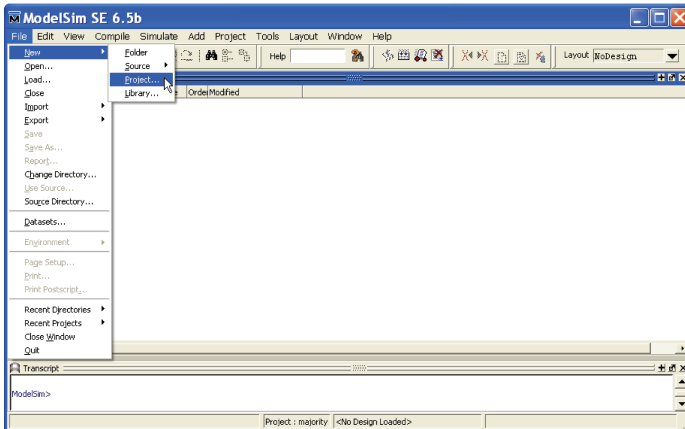


fig1

# Simulation VHDL

- Dans la fenêtre de la figure 2, cliquez sur browse → sélectionnez le chemin de GRT
- Entrez le nom du projet andcircuit
- OK

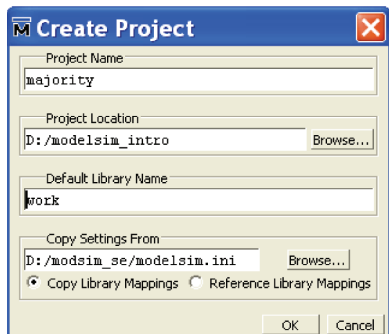


fig2

- Dans la fenêtre de la figure 3, cliquez sur Add Existing File

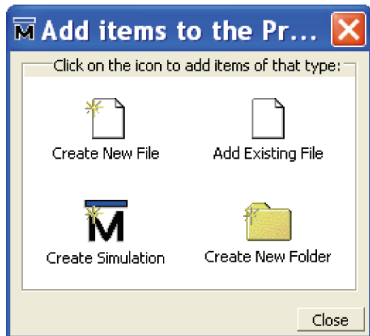


fig3



# Simulation VHDL

- Ajoutez le premier projet comme le montre la figure 4
- Refaites la même chose pour le deuxième projet

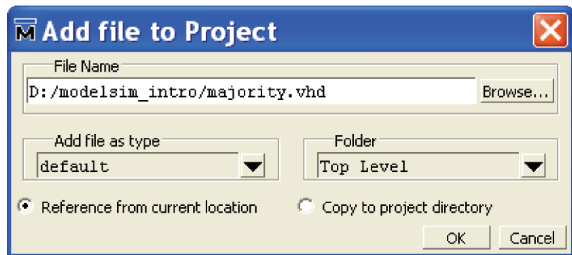


fig4

# Simulation VHDL

- La fenêtre principale du modelsim contient le projet (fig5)
- Remarquer le "?"

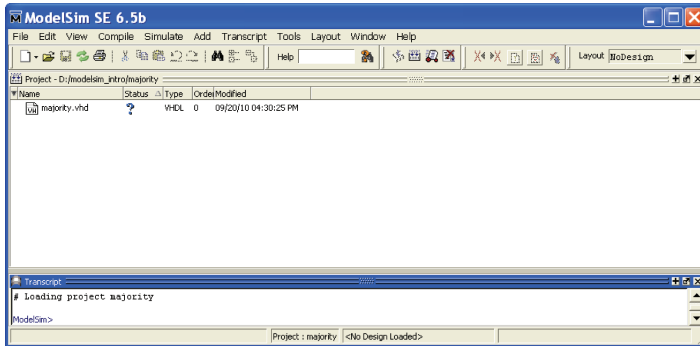


fig5

# Simulation VHDL

- Sélectionnez Compile → compile All
- Les projets sont compilés avec succès ✓ (fig6)

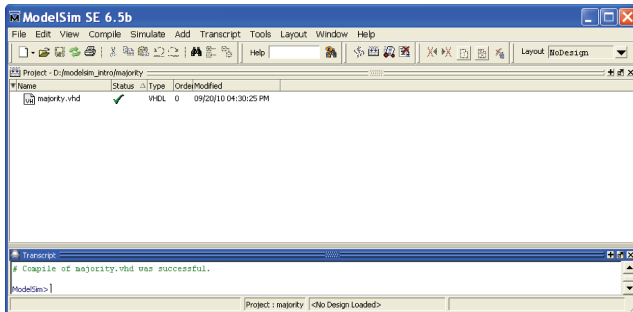


fig6

# Simulation VHDL

Création des signaux pour la simulation:

- Simulate → Start Simulation
- Vous aurez la fenêtre de la figure 7
- Ouvrez Work → sélectionnez andcircuit\_test puis OK

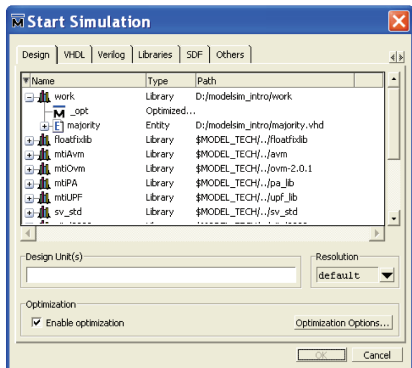
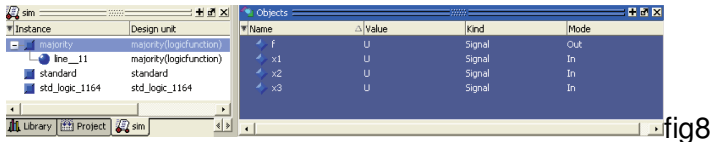


fig7

# Simulation VHDL

Création des signaux pour la simulation:

- Apparition de la fenêtre de la figure 8
- Contient les signaux d'entrée x et y, et de sortie z du circuit and
- Sélectionnez ces signaux pour les voir graphiquement



# Simulation VHDL

## Visualisation:

- View → Wave
- La fenêtre de la figure 9 apparaît
- Suivre la suite (Zoom, lancement de la simulation, arrêt ...)

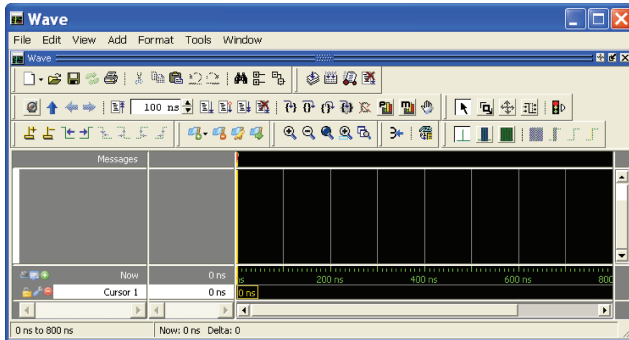


fig9

# Simulation VHDL

A refaire pour:

- OR
- $Z = \overline{A}.B + A.C$
- Multiplexeur 4-1 avec une entrée select (s)
- Encodeur 4-2 avec une sortie valid

Entrées				Sorties		
$A_3$	$A_2$	$A_1$	$A_0$	$Y_1$	$Y_0$	valid
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	1	0	0	1	0	1
1	0	0	0	1	1	1

- Convertisseur binaire-Gray (4 bits)

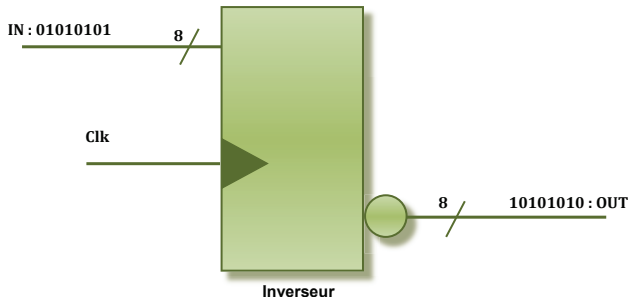
# Vérification d'un modèle HDL sous Simulink

- ❑ Le but de ce TP est de créer une plateforme de simulation en exploitant le bloc «**HDL Cosimulation**» de **Simulink**.
- ❑ Ce bloc permet de tester un composant matériel en lui appliquant des signaux d'entrée qui imitent les données réelles.
- ❑ La capture des signaux de sortie est assurée par le logiciel de simulation **ModelSim**.
- ❑ Le bloc «**HDL Cosimulation**» permet de tester des modèles conçus en **VHDL** ou en **Verilog**. Dans ce TP, le langage **VHDL** est utilisé.



## L'élaboration du code VHDL

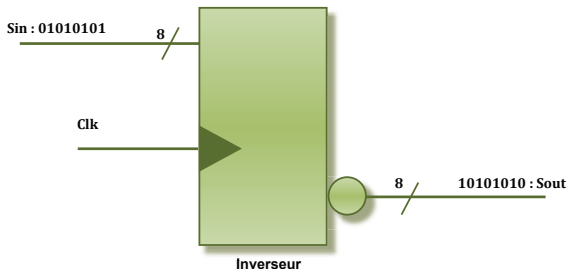
- ❑ Un scénario typique de l'utilisation de **Simulink** et de **ModelSim** consiste à créer dans ce dernier le modèle d'un composant matériel qui sera plus tard intégré dans **Simulink**.



# Vérification d'un modèle HDL sous Simulink

## L'élaboration du code VHDL

- ❑ L'entité VHDL de ce modèle est représentée sur 8-bit les deux ports **Sin** et **Sout** de type **STD\_LOGIC\_VECTOR**.
- ❑ Un signal d'horloge d'entrée de type **STD\_LOGIC** permet de déclencher le processus d'inversion (**clk**).



## L'élaboration du code VHDL

Effectuer les étapes suivantes:

- a. Démarrer **ModelSim**.
- b. Créer un répertoire **C:\MyPlayArea**.
- c. Dans l'invite de commande de **ModelSim>** taper la commande suivante : **cd C:/MyPlayArea**.
- d. Ouvrir un nouveau fichier VHDL (fenêtre d'édition).
- e. Coller dans ce nouveau fichier le code VHDL suivant:
- f. Enregistrer le fichier sous le nom: **inverter.vhd**.

# Vérification d'un modèle HDL sous Simulink

## L'élaboration du code VHDL

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY inverter IS PORT (
    Sin : IN std_logic_vector(7 DOWNT0 0);
    Sout: OUT std_logic_vector(7 DOWNT0 0);
    clk : IN std_logic
);
END inverter;

ARCHITECTURE behavioral OF inverter IS
BEGIN
    PROCESS(clk)
    BEGIN
        IF (clk'EVENT AND clk = '1') THEN
            Sout <= NOT Sin;
        END IF;
    END PROCESS;
END behavioral;
```

# Vérification d'un modèle HDL sous Simulink

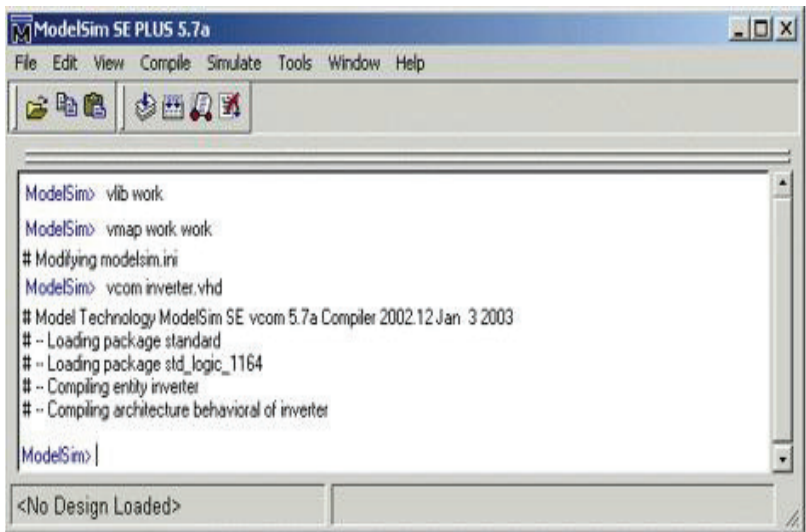
## Compilation du fichier VHDL

- ❑ Cette partie met en évidence la création d'une bibliothèque de conception, ainsi que la compilation du fichier **inverter.vhd**.
- ❑ Vérifier que le fichier **inverter.vhd** est dans le répertoire courant en tapant sur l'invite de commande de ModelSim la commande **dir**.
- ❑ Créer une bibliothèque de conception pour organiser les résultats de votre compilation en utilisant les commandes **Vlib** et **vmap** comme suit:  
**ModelSim> vlib work**  
**ModelSim> vmap work work**
- ❑ Pour compiler le fichier VHDL, sélectionner **Compile > Compile All**. Une autre façon pour compiler ce fichier, consiste à exécuter la commande **vcom**, comme suit:

**ModelSim> vcom inverter.vhd**

# Vérification d'un modèle HDL sous Simulink

## Compilation du fichier VHDL



```
ModelSim> vlib work
ModelSim> vmap work work
# Modifying modelsim.ini
ModelSim> vcom inverter.vhd
# Model Technology ModelSim SE vcom 5.7a Compiler 2002.12 Jan 3 2003
# -- Loading package standard
# -- Loading package std_logic_1164
# -- Compiling entity inverter
# -- Compiling architecture behavioral of inverter
ModelSim> |
```

<No Design Loaded>

# Vérification d'un modèle HDL sous Simulink

## Création du modèle Simulink

- ☐ Démarrer **MATLAB** puis **Simulink**.
- ☐ Créer une nouvelle fenêtre de travail.
- ☐ Glisser les blocs suivants à partir des Bibliothèques **Simulink** dans la fenêtre de travail:
  - **Constant** de la bibliothèque **Source**.
  - **HDL cosimulation** de la bibliothèque **EDA Simulator Link MQ**.
  - **Display** de la bibliothèque **Sink**.
- ☐ Placer les trois blocs dans l'ordre indiqué sur la figure suivante :



# Vérification d'un modèle HDL sous Simulink

## Création du modèle Simulink

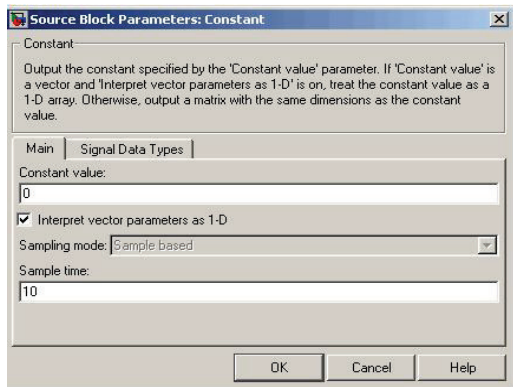
- ❑ Configurer le bloc **Constant** en effectuant les actions suivantes:
- ❑ Pour ouvrir la boîte de dialogue des paramètres, double-cliquer sur **Constant**.

Entrer les paramètres suivants :

**Constant value:** 0

**Sample time:** 10

La boîte de dialogue doit apparaître comme suit :

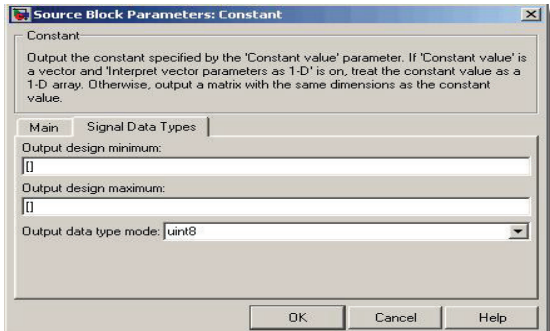




# Vérification d'un modèle HDL sous Simulink

## Création du modèle Simulink

- ☐ Cliquer sur l'onglet « **Signal data types** ». La boîte de dialogue affiche le type de donnée de sortie.
- ☐ Sélectionner « **uint8** ». Cette configuration correspond directement au type de variable utilisé dans le code VHDL :  
**Sin : IN std\_logic\_vector(7 DOWNT0 0)**
- ☐ La boîte de dialogue doit apparaître comme suit :



- ☐ Cliquer sur OK

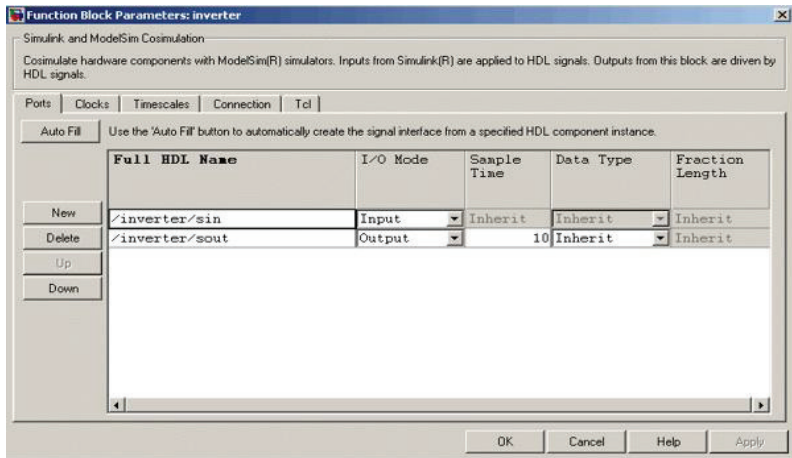
## Création du modèle Simulink

- ❑ configurer le bloc «**HDL cosimulation**» en effectuant les actions suivantes:
  1. Double-cliquer sur le bloc «**HDL cosimulation**».
  2. Dans l'onglet **Ports**, sélectionnez le signal **/top/sig1**.
  3. Remplacer le signal **/top/sig1** par **/inverter/Sin**.
  4. Remplacer le signal **/top/sig2** par **/inverter/Sout**. Puis sélectionner «**Output**» sur le paramètre «**I/O Mode**». Changer la valeur du paramètre «**Sample Time**» par **10**.
  5. Sélectionner le signal **/top/sig3**. Cliquez sur le bouton **Delete**.
  6. Cliquez sur **Apply**.

# Vérification d'un modèle HDL sous Simulink

## Création du modèle Simulink

L'onglet ports doit apparaître comme suit :



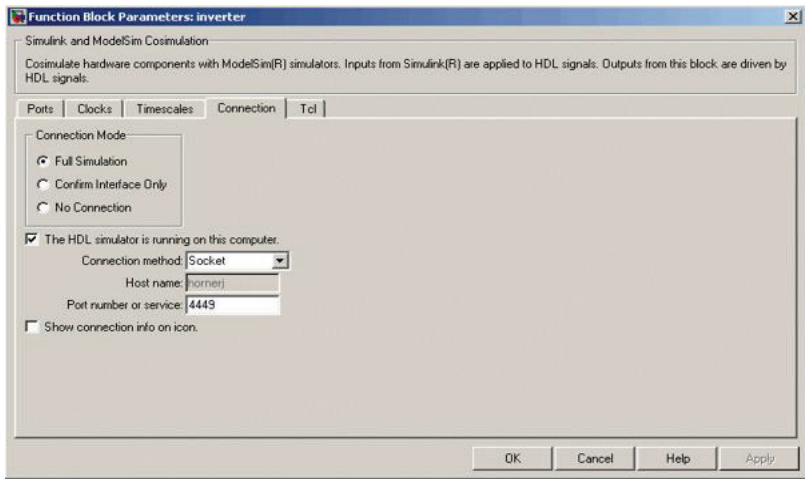
## Création du modèle Simulink

- ❑ Configurer les paramètres de l'onglet «**Connection**» en effectuant les actions suivantes:
  1. Cliquer sur l'onglet «**Connection**».
  2. Garder «**Connection Mode**» sur «**Full Simulation**».
  3. Sélectionner « **socket** » dans la liste « **Connection method** ». Cette option spécifie que Simulink et ModelSim seront en communication via une liaison TCP / IP socket.
  4. Dans la zone de texte du numéro de port, entrer la valeur **4449**.
  5. Cliquer sur **Apply**.

# Vérification d'un modèle HDL sous Simulink

## Création du modèle Simulink

L'onglet «**Connection**» doit apparaître comme suit :



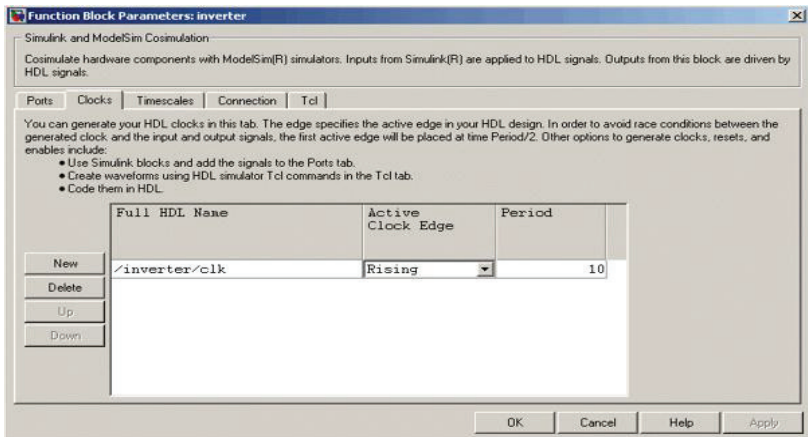
## Création du modèle Simulink

- ❑ Configurer l'onglet «**Clocks**» en effectuant les actions suivantes:
  1. Cliquer sur l'onglet «**Clocks**».
  2. Cliquer sur le bouton «**New**».
  3. Double-cliquer sur le nom du nouveau signal et entrer le chemin du signal suivant : **/inverter/clk**.
  4. Sélectionner «**Rising**» dans la liste «**Edge**».
  5. Entrer la valeur **10** dans le champ «**Period**».
  6. Cliquer sur **Apply**

# Vérification d'un modèle HDL sous Simulink

## Création du modèle Simulink

L'onglet «**Clocks**» doit apparaître comme suit :



## Création du modèle Simulink

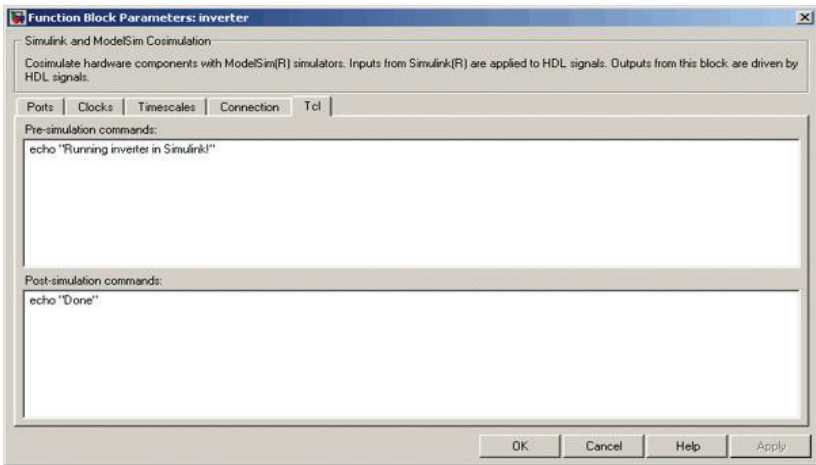
- ❑ Configurer l'onglet «**Tcl**» en effectuant les actions suivantes:
  1. Cliquer sur l'onglet **Tcl**.
  2. Dans la zone «**pré-simulation text commandes**», entrer la commande suivante:  
**echo "Running inverter in Simulink!"**
  3. Dans la zone «**post-simulation commandes**», entrer la commande suivante:  
**echo "Done"**



# Vérification d'un modèle HDL sous Simulink

## Création du modèle Simulink

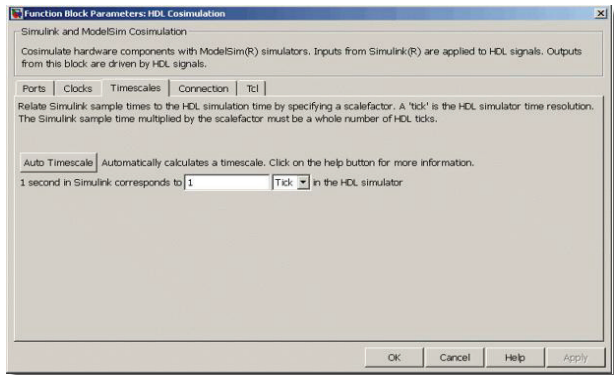
L'onglet «Tcl» doit apparaître comme suit :



# Vérification d'un modèle HDL sous Simulink

## Création du modèle Simulink

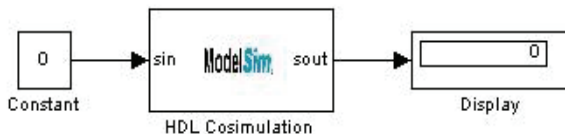
- ❑ Afficher l'onglet «**Timescales**» pour s'assurer que les paramètres sont réglés comme suit:



- ❑ Cliquer sur OK pour fermer la boîte de dialogue.

## Création du modèle Simulink

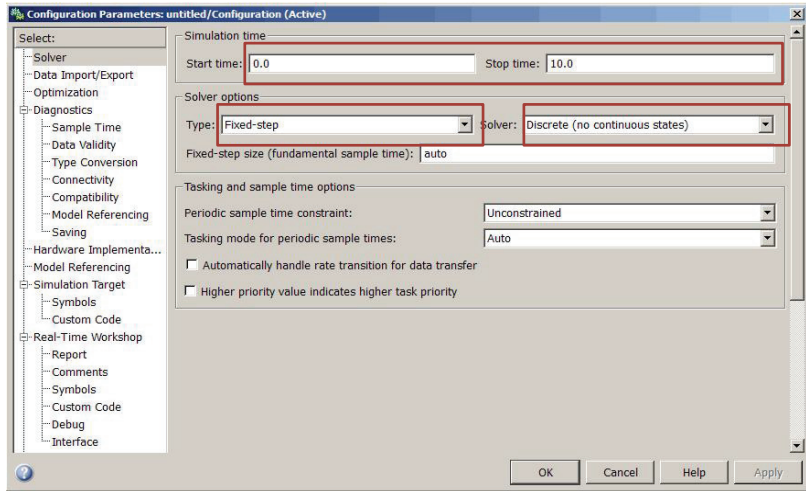
- ❑ Connecter les blocs comme illustré sur la figure suivante:



- ❑ Sélectionner «**Configuration Parameters**» dans le menu « **Simulation** » de Simulink.
- ❑ Configurer les paramètres comme indiqué sur la figure suivante :

# Vérification d'un modèle HDL sous Simulink

## Création du modèle Simulink



## Etablir la connexion entre ModelSim et Simulink

- ❑ Vous disposez maintenant d'une description VHDL de l'inverseur et d'un modèle Simulink implémentant cette description.
- ❑ Avant de continuer, fermer **ModelSim**
- ❑ Pour établir une connexion de type socket entre **ModelSim** et **Simulink**, entrer la ligne de commande suivante dans l'invite de commande de **MATLAB**:

```
vsim('socketsimulink', 4449)
```

- ❑ Dans l'invite de commande de **ModelSim**, exécuter les commandes suivantes :

```
ModelSim> cd C:/MyPlayArea
```

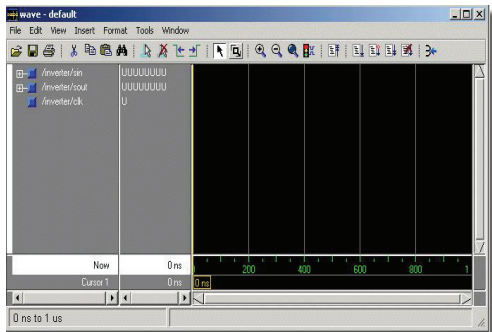
```
ModelSim> vsimulink work.inverter
```

# Vérification d'un modèle HDL sous Simulink

## Lancer la simulation

- ❑ Pour lancer la simulation, effectuer les actions suivantes:
  1. Ajouter les signaux de l'inverseur à la fenêtre «**wave**» en exécutant sur l'invite de commande de **ModelSim** la commande suivante :

**VSIM n> add wave /inverter/\***



1. Sur Simulink, lancer la simulation en cliquant sur le bouton «**Run**».

