

VHDL

CONFIGURATION

- Plusieurs architectures peuvent être associées à une entité
- Le rôle principal d'une configuration est de préciser cette association

```
configuration demo_config of even_detector_tb is
  for tb_arch
    for uut: even_detector
      use entity work.even_detector(sop_archi);
    end for;
  end for;
end demo_config;
```

- Une configuration directe dans le corps de l'architecture (en précisant la `library.composant`)

SOMMAIRE

- 3 Utilisation de Quartus
 - Création du projet Quartus
 - Utilisation du flot de conception

CRÉATION DU PROJET QUARTUS

UTILISATION DU PROJECT WIZARD

- page 1 : Donner un nom au projet et choisir un dossier
- page 2 : Ne rien changer
- page 3 : Spécifions le FPGA de la carte de développement :
 - ▷ Family Cyclone V
 - ▷ Devices Cyclone V SE Mainstream
 - ▷ Target device ⇒ Specific device
 - ▷ Sélectionner 5CSEMA5F31C6
- page 4 : Ne rien changer
- page 5 : Cliquer sur Finish

Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

Device family

Family: Cyclone V (E/GX/GT/SX/SE/ST)

Devices: Cyclone V SE Mainstream

Target device

☐ Auto device selected by the Fitter

☒ Specific device selected in 'Available devices' list

☐ Other: n/a

Show in 'Available devices' list:

Package: Any

Pin count: Any

Speed grade: Any

Name filter:

☒ Show advanced devices

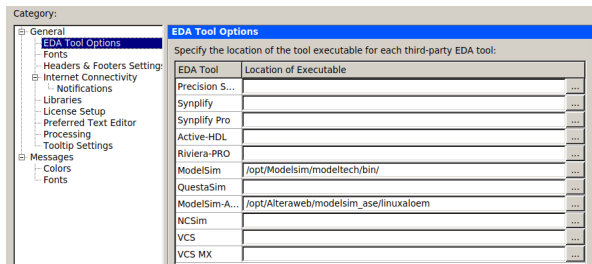
Available devices:

Name	Core Voltage	ALMs	User I/Os	GXB Channel PM
5CSEMA4U2317 (Advanced)	1.1V	15880	314	0
5CSEMA5F31A7	1.1V	32070	437	0
5CSEMA5F31C6	1.1V	32070	437	0

CONFIGURATION DE QUARTUS

CHOIX DE LA VERSION DE MODELSIM

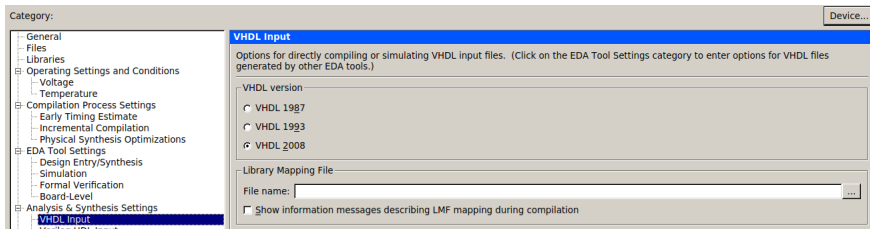
- Si l'on ne possède pas de licence pour modelsim :
- Menu **tools** ⇒ **Options** ⇒ pointer vers la version Altera Starter Edition



CONFIGURATION DE QUARTUS

CHOIX DE LA VERSION DE VHDL

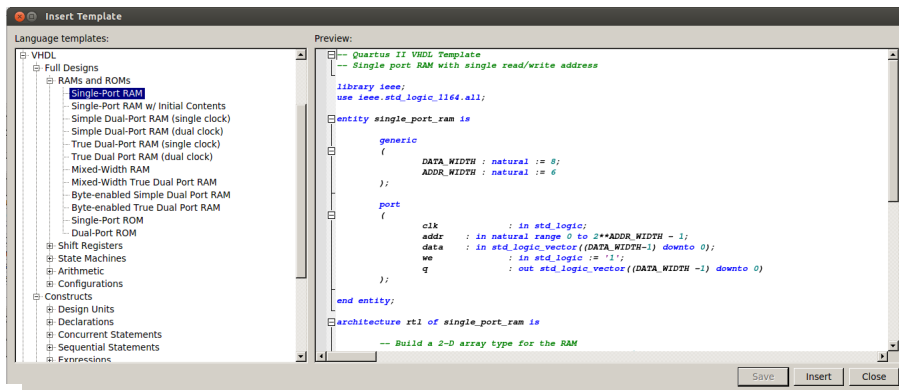
- Pour bénéficier des facilités offertes par VHDL-2008
 - ▷ Clic droit sur device dans Hierarchy ⇒ Settings
 - ▷ VHDL input ⇒ cocher VHDL-2008
 - ▷ Ok



EDITION DU CODE VHDL

UTILISATION DES MODÈLES

- Quartus offre de nombreux modèles de codage VHDL
 - ▷ Edit ⇒ Insert Template
 - ▷ Choisir le modèle souhaité
 - ▷ Insert



SOMMAIRE

- ③ Utilisation de Quartus
 - Création du projet Quartus
 - Utilisation du flot de conception

EXEMPLE TRIVIAL : PORTE OU-EXCLUSIF

CODE VHDL

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

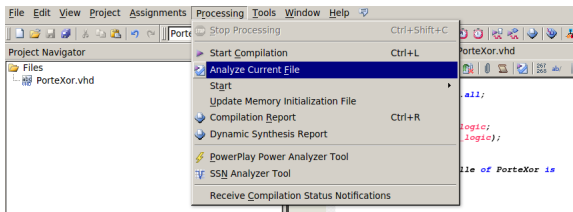
entity PorteXor is
    port( A, B : in std_logic;
          S : out std_logic);
end PorteXor;

architecture fonctionnelle of PorteXor is
begin
    S <= A xor B;
end fonctionnelle;
```


EXEMPLE TRIVIAL : PORTE OU-EXCLUSIF

FLOT DE CONCEPTION

- Vérifier que le code ne comporte pas d'erreurs de syntaxe
 - ▷ Processing ⇒ Analyze current file

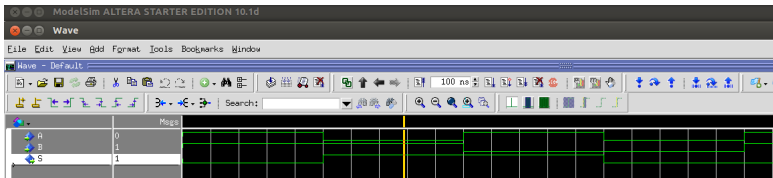


- Pas d'erreurs ⇒ Compiler (Ctrl-L) puis démarrer le simulateur
 - ▷ Tools ⇒ Run Simulation Tool ⇒ RTL Simulation

EXEMPLE TRIVIAL : PORTE OU-EXCLUSIF

MODELSIM

- ❑ Sélectionner l'entité `portexor` de la bibliothèque `work` (double clic)
- ❑ Dans la fenêtre `Objects` sélectionner les signaux intéressants
- ❑ Les ajouter aux chronogrammes (clic droit \Rightarrow add wave)
- ❑ Les modifier (clic droit \Rightarrow modify \Rightarrow apply clock ou apply wave)
- ❑ Démarrer la simulation : `run 300 ns`



EXEMPLE TRIVIAL : PORTE OU-EXCLUSIF

MODELSIM

- Pour ne pas avoir à refaire la configuration graphique de **modelsim** à chaque itération, enregistrer la configuration :
 - ▷ Soit l'ensemble par la commande : `write transcript msimxor.do`
 - ▷ Soit la configuration de la fenêtre **wave** uniquement : `file`
⇒ `save format` qui créera un fichier `wave.do`
- Lors d'une nouvelle ouverture de **modelsim** taper l'une des commandes
 - ▷ `do msimxor.do`
 - ▷ `do wave.do`
- Démarrer la simulation : `run 300 ns`
- En pratique on préfère utiliser un **testbench** écrit directement en VHDL

EXEMPLE TRIVIAL : PORTE OU-EXCLUSIF

TESTBENCH

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

entity PorteXor_tb is
end PorteXor_tb;

architecture tb of PorteXor_tb is
    --passage de l'entité PorteXor au testbench comme
    --composant
    component PorteXor is
    port( A, B : in std_logic;
          S : out std_logic);
    end component;
```

EXEMPLE TRIVIAL : PORTE OU-EXCLUSIF

TESTBENCH

```
signal inA, inB, outS : std_logic;
begin
  --relier les signaux du testbench aux ports de
  PorteXor
  mapping: PorteXor port map(inA, inB, outS);

  process
    --variable pour les erreurs
    variable errCnt : integer := 0;
  begin
    --TEST 1
    inA <= '0';
    inB <= '0';
    wait for 15 ns;
```

EXEMPLE TRIVIAL : PORTE OU-EXCLUSIF

TESTBENCH

```
    assert(outS = '0')    report "Error 1"    severity
error;
    if(outS /= '0') then
        errCnt := errCnt + 1;
    end if;

--TEST 2
inA <= '0';
inB <= '1';
wait for 15 ns;
    assert(outS = '1')    report "Error 2"    severity
error;
    if(outS /= '1') then
        errCnt := errCnt + 1;
```

EXEMPLE TRIVIAL : PORTE OU-EXCLUSIF

TESTBENCH

```
end if;

--TEST 3
inA <= '1';
inB <= '1';
wait for 15 ns;
assert(outS = '0') report "Error 3" severity
error;
if(outS /= '0') then
    errCnt := errCnt + 1;
end if;

----- RESUME -----
if(errCnt = 0) then
```

EXEMPLE TRIVIAL : PORTE OU-EXCLUSIF

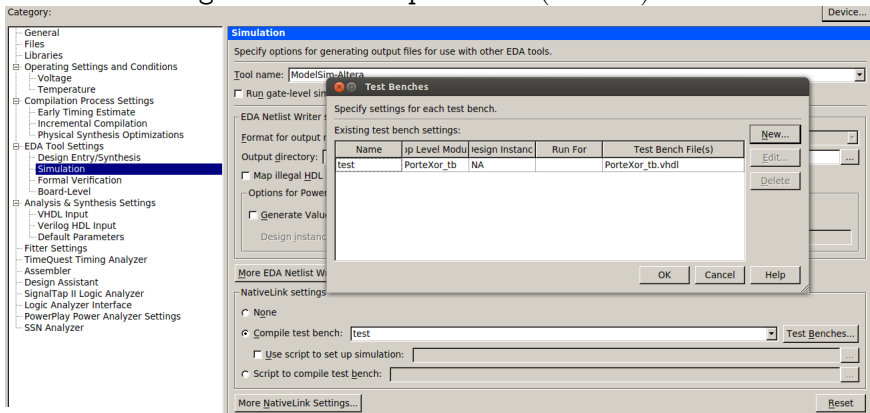
TESTBENCH

```
        assert false report "OK!"    severity note;  
    else  
        assert true report "KO!"     severity error;  
    end if;  
  
    end process;  
end tb;
```


EXEMPLE TRIVIAL : PORTE OU-EXCLUSIF

TESTBENCH

- Spécification du fichier testbench dans les settings du projet Quartus
- Puis compilation de l'ensemble système + testbench :
Processing ⇒ Start compilation (Ctrl-L)



EXEMPLE TRIVIAL : PORTE OU-EXCLUSIF

TESTBENCH

- Quartus peut aussi générer un modèle de testbench :
Processing \Rightarrow Start \Rightarrow Start Testbench Template Writer
- Cela crée un fichier .vht (dans le dossier modelsim) qu'il faut compléter puis spécifier en tant que testbench comme vu précédemment
- Dans tous les cas on doit obtenir un résultat de simulation du type suivant :

```
run 100 ns
# ** Note: OK!
#   Time: 45 ns   Iteration: 0   Instance: /portexor_tb
# ** Note: OK!
#   Time: 90 ns   Iteration: 0   Instance: /portexor_tb
```