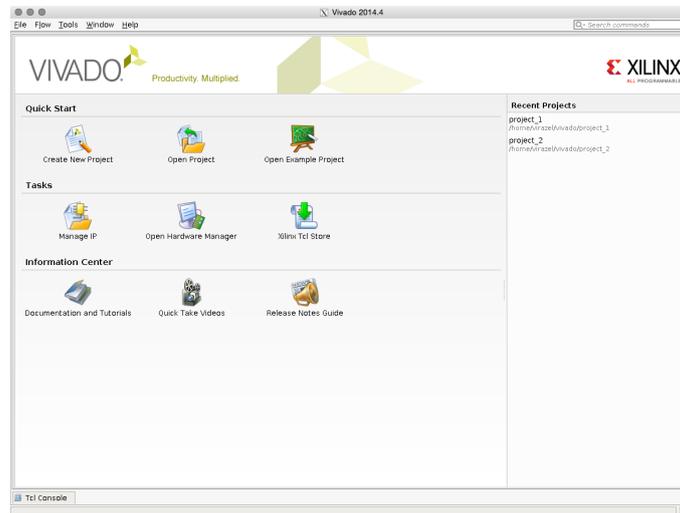


TP VHDL N°1

L'outil Xilinx VIVADO

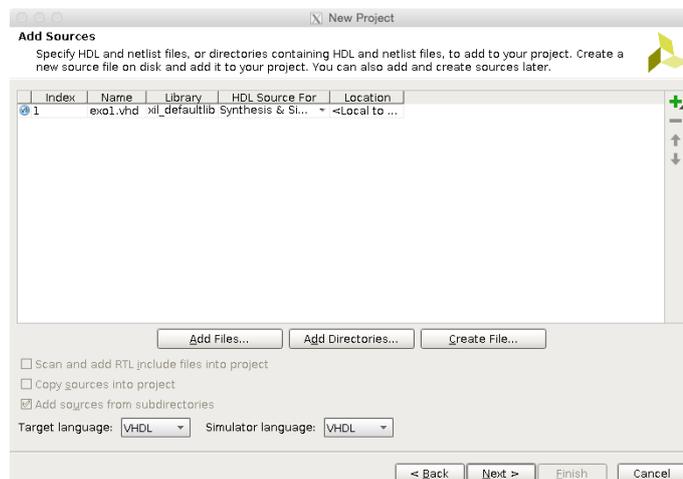
Lancement

Lancer VIVADO



Créer un nouveau projet

1. Cliquez sur « *Create New Project* ».
2. Cliquez sur « *Next* ».
3. Choisir le nom du projet « *Project name* » ainsi que son emplacement de sauvegarde « *Project Location* ».
4. Cliquez sur « *Next* ».
5. Sélectionner le « *Project Type* » : *RTL Project*.
6. Cliquez sur « *Next* ».
7. Cliquez sur « *Create File* » afin de créer le fichier qui contiendra la description VHDL que vous aller réaliser.
8. Dans la fenêtre indiquer le nom du fichier sans l'extension puis cliquez sur « *Ok* ».

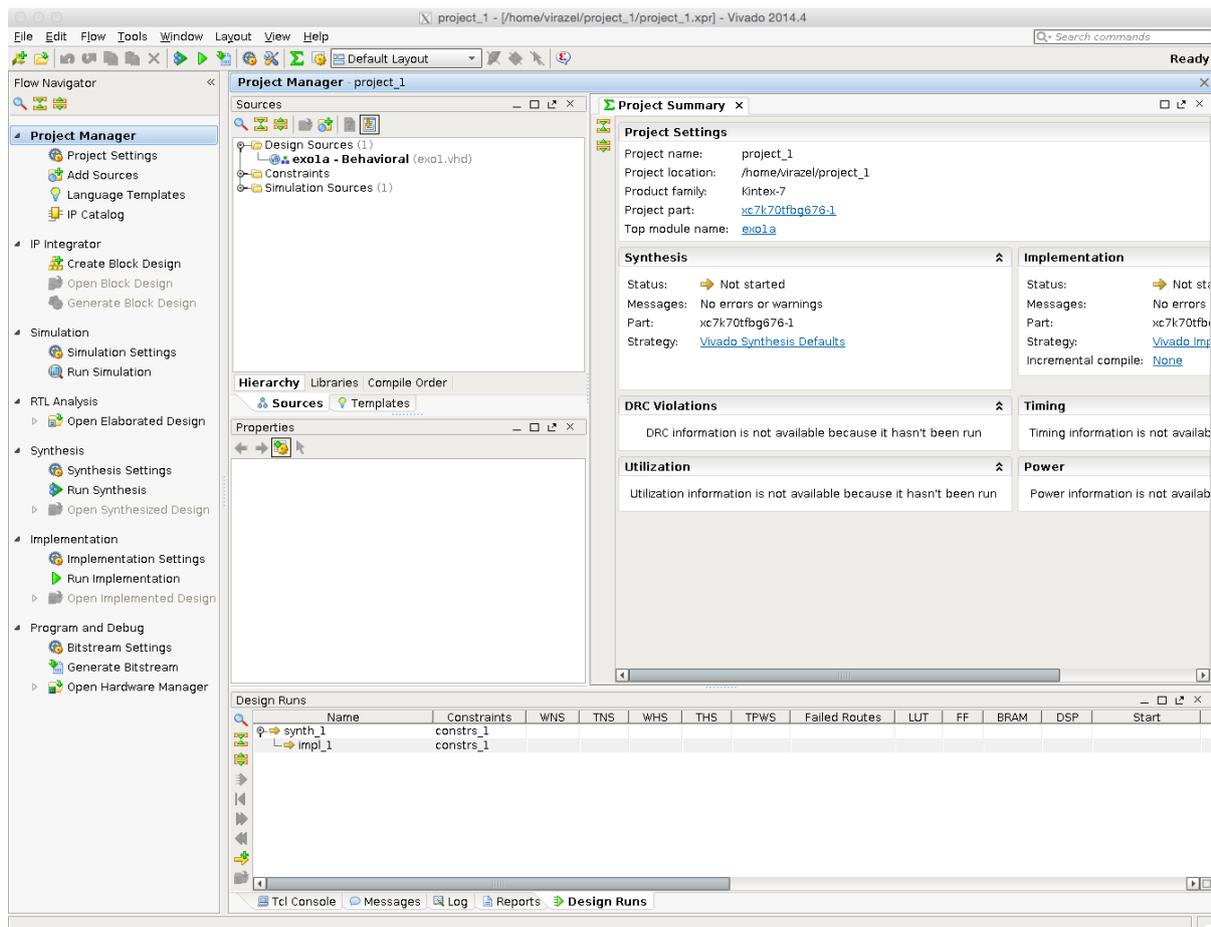


9. Cliquez trois fois sur « *Next* ».

10. Vous devez maintenant sélectionner la carte/FPGA cible. Laissez pour l'instant la sélection par défaut puis cliquez sur « *Next* ».
11. La fenêtre « *New Project Summary* » vous indique le bilan de la configuration. Cliquez sur « *Finish* ».

Créer une source HDL

12. Vous devez maintenant définir les entrées et sorties de l'« *entity* » permettant de créer un MUX à 2 entrées.
13. Cliquez sur « *Ok* »
14. Vous devez maintenant avoir la fenêtre suivante :



15. Complétez l'architecture afin de réaliser la fonction MUX demandée.

Simulation

16. Il faut maintenant créer un *testbench* afin de simuler votre description. Dans le menu « *Files* » sélectionner « *Add source* ».
17. Sélectionner « *Add or Create Simulation Source* » puis cliquez sur « *Next* »
18. Cliquez sur « *Create File* » et nommez votre fichier `exo1_tb` puis cliquez sur « *Finish* ».
19. Dans la fenêtre permettant de créer les entrées et sorties, ne rien rajouter (un *testbench* est dépourvu d'entrée/sortie) puis cliquez sur « *Ok* ».
20. Editer le fichier *testbench* afin de décrire la simulation que vous souhaitez lancer

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity ex01_tb is
end ex01_tb;

architecture Behavioral of ex01_tb is
  -- Component Declaration
  component ex01a
  port(
    A : in STD_LOGIC;
    B : in STD_LOGIC;
    COM : in STD_LOGIC;
    S : OUT STD_LOGIC);
  end component;

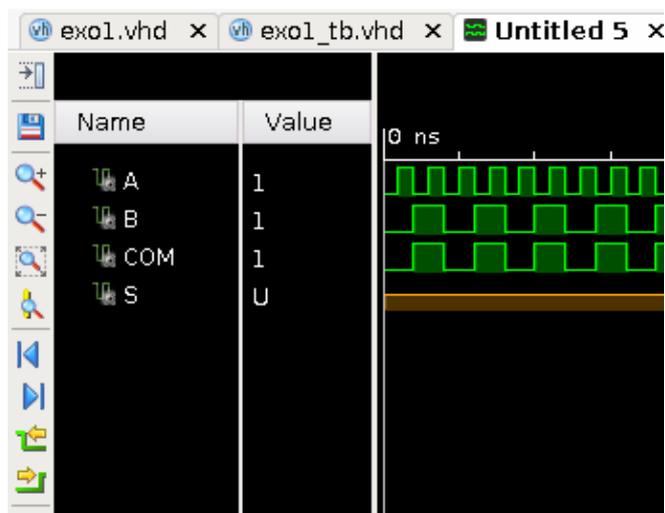
  signal A, B, COM, S : STD_LOGIC;

begin
  -- Component Instantiation
  c1 : ex01a port map(A, B, COM, S);

  tb : process
  begin
    A <= '0';
    B <= '0';
    COM <= '0';
    wait for 20ns;
    A <= '1';
    B <= '0';
    COM <= '0';
    wait for 20ns;
    A <= '0';
    B <= '1';
    COM <= '1';
    wait for 20ns;
    A <= '1';
    B <= '1';
    COM <= '1';
    wait for 20ns;
  end process tb;
end Behavioral;

```

21. Afin de simuler le *testbench* il faut le définir comme un « *top module* ». Clic droit sur le fichier et sélectionner « *Set as Top* ».
22. Cliquez sur « *Run Simulation* » et Sélectionnez « *Run Behavioral Simulation* ». Vous devez obtenir la fenêtre de simulation suivante. Utiliser les options de zoom pour ajuster l’affichage et vérifier votre simulation.



Questions – Logique Combinatoire

Comparateur

- Réaliser et simuler une fonction réalisant la comparaison de deux nombres binaires de 4 bits. Trois sorties (actives à l'état haut) permettent de fournir le résultat de la comparaison :
 - une sortie pour « supérieur »
 - une sortie pour « inférieur »
 - une sortie pour « égal »

Additionneurs

- Réaliser et simuler un additionneur 8 bits avec retenue sortante. Réalisez cet additionneur de deux façons :
 - comportementale (en utilisant les bibliothèques *IEEE.std_logic_arith.all* et *IEEE.std_logic_unsigned.all*)
 - structurelle (en utilisant un bloc full adder 1 bit).
- Réaliser et simuler un additionneur/soustracteur (si com = 1 addition, si com = 0 soustraction) 8 bits avec retenue sortante.

Les Opérateurs et Délimiteurs

Délimiteurs	Nom	Exemple
+	Addition et positif	Somme := +5 + 10;
-	Soustraction et négatif	Différence := -7 - 9;
*	Multiplication	Mult := 7 * 9;
/	Division	Div := 10 / 2;
=	Égalité (comparaison)	If valeur = 5 then
<	Plus petit que	If somme < 6 then
>	Plus grand que	If diff > 7 then
&	Concaténation	Quatre_bits := Deux_bits & "10";
	Barre verticale (choix multiple)	When 'Y' 'y'
;	terminateur	Fin d'instruction
#	Délimiteur sz base	X := 16#AB34#; -- (en base 16)
()	Parenthèses (niveau de précedence)	Y := (5 + sum)*10;
:	Séparation d'un objet et de son type	Variable x : integer;
"	Délimiteur de chaîne	Report "message"
'	Séparateur d'attribut	X := Y'left;
**	Puissance	Z := 2 ** 3 -- égale 8.
=>	Dire "alors"	When val_1 => X := 5;
=>	Dire "prend"	Tableau := (Element1 => 5, autres => 100);
:=	Assignement d'une variable	X := 4 + 5;
/=	Inégalité (différent)	If x /= 5 then
>=	Plus grand ou égal	If x >= 3 then
<=	Plus petit ou égal	If x <= 3 then
<=	Assignement d'un signal	Sortie <= 1 after 10 ns;
--	Ligne de commentaires	