Projet RTL (VHDL) basique

Introduction

Dans ce tutoriel, nous allons créer un projet VHDL simple en utilisant l'éditeur de texte de Xilinx Vivado 2016.1.

Le design aura 4 entrées et 1 sortie de taille 1 bit. Les 2 premières entrées, que nous allons appeler A et B, seront connectée à un ET logique (AND) et les 2 autres, C et D, seront connectées à un OU logique (OR). Les sorties du ET et du OU logiques seront connectées à un autre ET logique. Ce projet est schématisé en Figure 1.

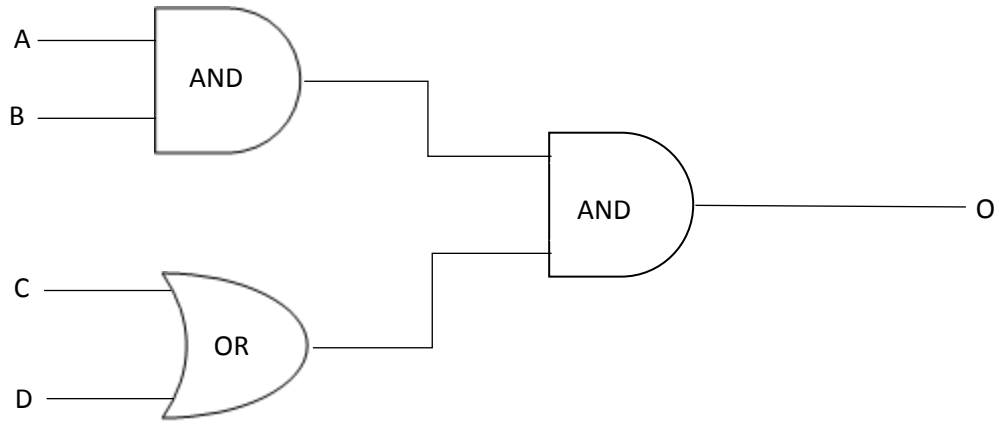

Figure 1 – Schématique du projet

Créer un projet VHDL

Démarrez Xilinx Vivado 2016.1 en mode graphique via l'invite de commande (comme expliqué dans le post 1) ou via l'icône du bureau. Dans la page d'accueil de Vivado, cliquez sur « *Create New Project* ». Entrez ensuite le nom du projet ainsi que son emplacement.

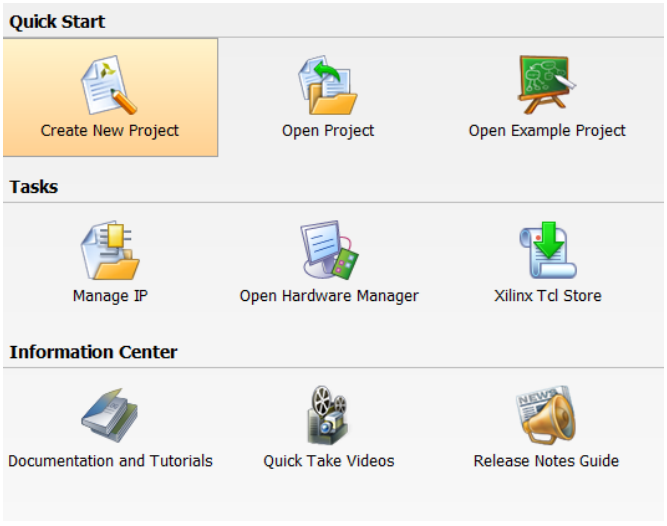

Figure 2 – Créer un nouveau projet

© Copyright 2016 www.dev-flow.com

Dans la fenêtre “Project Type”, cliquez sur “*RTL project*” et sélectionnez “*Do not specify sources at this time*”.

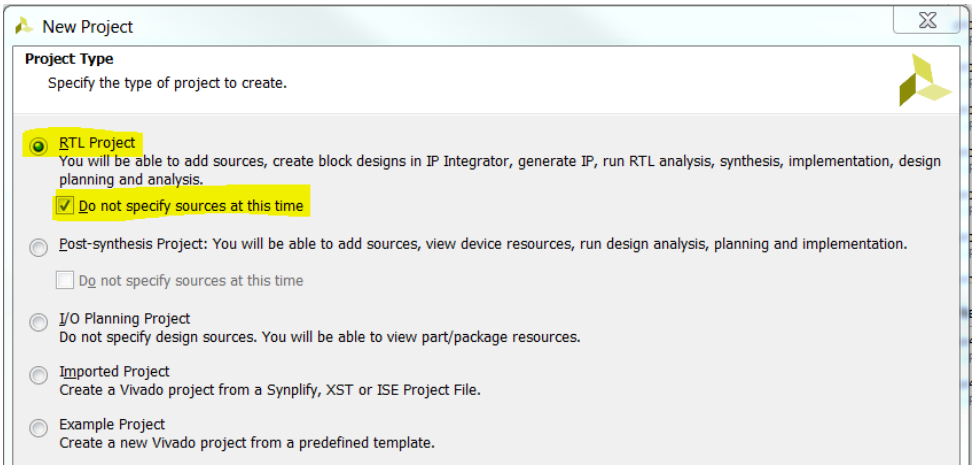

Figure 3 – Sélection du type de projet (RTL)

Sélectionnez le Kintex-7 xc7k70tfbg676-1 comme FPGA cible dans la fenêtre « *Default Part* ».

Créer le fichier VHDL

Dans le Flow Navigator, cliquez sur “*Add Sources*” (alternativement vous pouvez cliquer sur “*File* > *Add Sources*”).

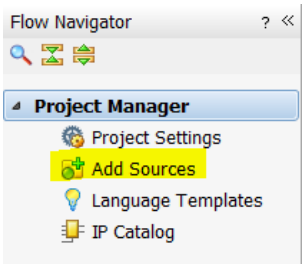

Figure 4 – Bouton « Add Sources » de Vivado

Dans la première page de la fenêtre “*Add Sources*”, sélectionnez “*Add or create design sources*”.

Figure 5 - Add or create design sources

© Copyright 2016 www.dev-flow.com

Dans la page "*Add or Create Design Sources*", cliquez sur le bouton + sur la gauche et cliquez sur "*Create File*" ou cliquez sur le bouton "*Create File*" situé en bas de la page.

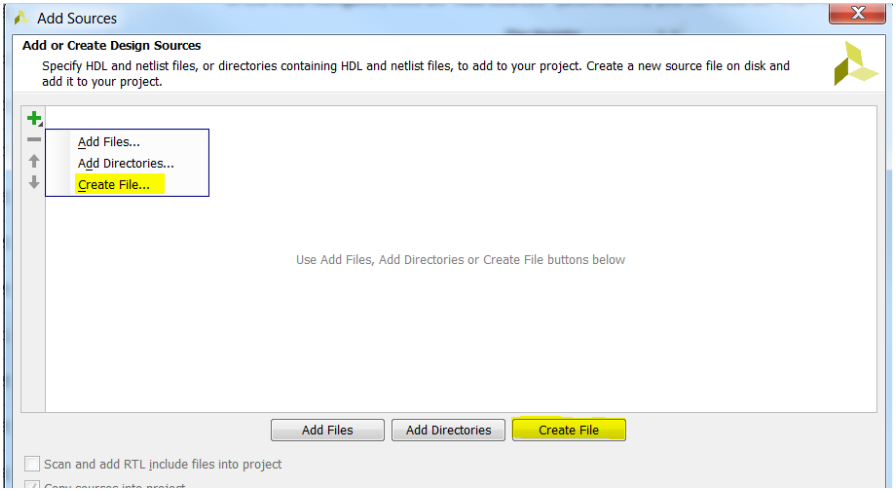

Figure 6 – Créer un nouveau fichier source

Dans la fenêtre "*Create Source File*", sélectionnez "*VHDL*" pour l'option "*File type*" et entrez le nom du fichier. Cliquez sur "*OK*" puis sur "*Finish*" dans la fenêtre "*Add Sources*".

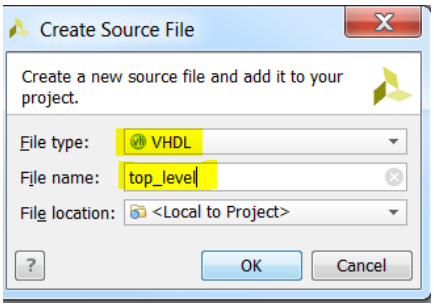

Figure 7 – Créer un nouveau fichier VHDL

Une fenêtre "*Define Module*" doit apparaitre. Elle permet de définir les ports de votre nouveau module. Un fichier VHDL pré-rempli sera alors créé. Remplissez la fenêtre telle que la Figure 8.

© Copyright 2016 www.dev-flow.com

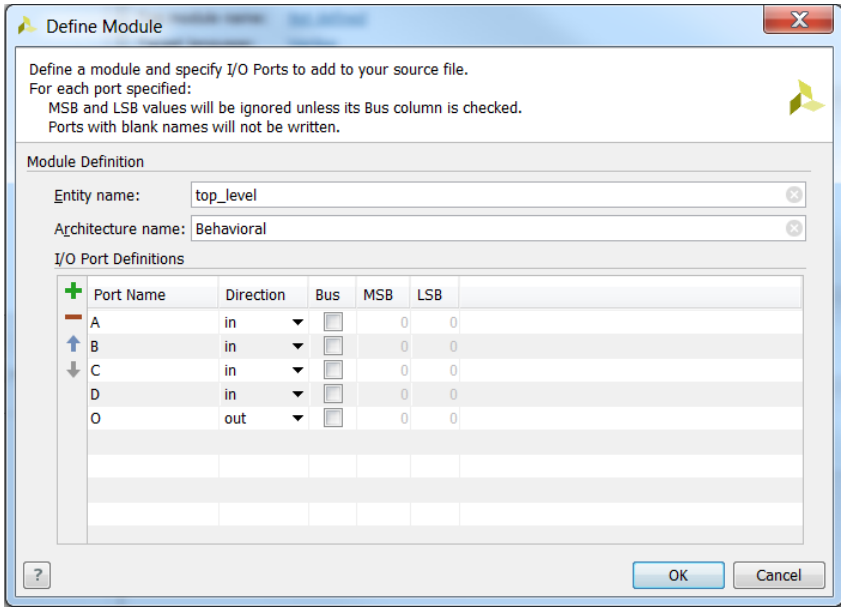

Figure 8 – Fenêtre « Define Module »

Modifier le fichier VHDL

Dans la fenêtre "*Sources*" de la zone « *Data Windows* », le fichier VHDL que nous avons créé apparait sous "*Design Sources*". Double-cliquez sur le fichier pour l'ouvrir dans l'éditeur de texte.

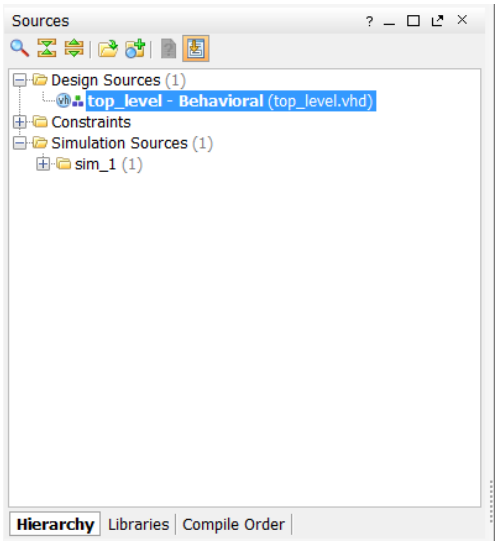

Figure 9 - Fenêtre Sources

Nous pouvons voir que Vivado a déjà créé la structure du fichier VHDL.

© Copyright 2016 www.dev-flow.com

```
34 entity top_level is
35     Port ( A : in STD_LOGIC;
36            B : in STD_LOGIC;
37            C : in STD_LOGIC;
38            D : in STD_LOGIC;
39            O : out STD_LOGIC);
40 end top_level;
41
42 architecture Behavioral of top_level is
43
44 begin
45
46
47 end Behavioral;
```

Figure 10 – Structure du fichier VHDL créée par Vivado

Dans l'éditeur de texte, nous pouvons coder notre bloc.

```
42 architecture Behavioral of top_level is
43
44 signal AND_1_out_sig  :  std_logic;
45 signal OR_out_sig    :  std_logic;
46
47 begin
48
49 AND_1_out_sig   <= A AND B;
50 OR_out_sig      <= C OR D;
51
52 O <= AND_1_out_sig AND OR_out_sig;
53
54
55 end Behavioral;
```

Figure 11 – Code VHDL de notre bloc

Quand vous sauvegarder votre fichier, si il y a des erreurs de syntaxe, le fichier sera affiché sous "*Syntax Error Files*" dans la fenêtre "*Sources*" et les erreurs seront affichées dans la console de message.

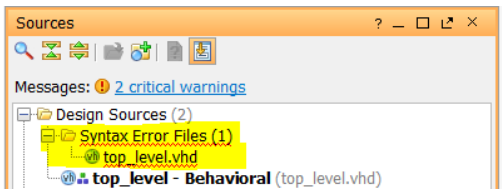

Figure 12 - Syntax Error Files

Ouvrir le schématique du bloc RTL (VHDL)

Dans le "*Flow Navigator*", sous "*RTL Analysis*", si l'on étend "*Elaborated Design*" on peut cliquer sur "*Schematic*".

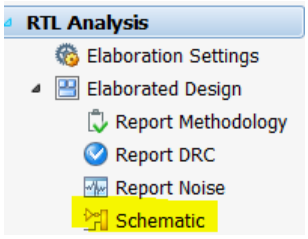

Figure 13 – Ouvrir le Schématique élaboré du design

© Copyright 2016 www.dev-flow.com

Cela permet d'afficher le schématique de notre bloc RTL. Ce schématique nous permet de vérifier que nous avons codé notre bloc correctement.

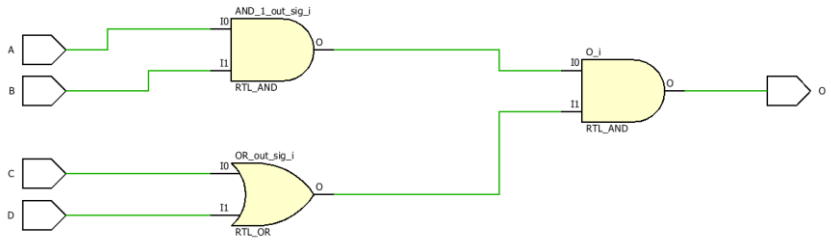

Figure 14 – Schématique élaboré du design

© Copyright 2016 www.dev-flow.com

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity top_level is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : in STD_LOGIC;
           D : in STD_LOGIC;
           O : out STD_LOGIC);
end top_level;

architecture Behavioral of top_level is

signal AND_1_out_sig    : std_logic;
signal OR_out_sig       : std_logic;

begin

AND_1_out_sig   <= A AND B;
OR_out_sig      <= C OR D;

O   <= AND_1_out_sig AND OR_out_sig;

end Behavioral;
```

© Copyright 2016 www.dev-flow.com