**Licence SPI – Sciences 3 SNI 3 (VHDL 1/FPGA) - 14/11/2024**



**4 Teams Project 1 :**

**Détecteur de Parité – VHDL/FPGA 1 – Découverte GitHub**

Table des matières

[*I.* *Description du projet :* 3](#_Toc182481261)

[*II.* *Détail des différentes parties pour le code VHDL :* 3](#_Toc182481262)

[*III.* *Présentation des équipes :* 3](#_Toc182481263)

[*IV.* *Application VIVADO* 4](#_Toc182481264)

[*Equipe 1 : Calcul de la Parité pour les Bits Pairs* 4](#_Toc182481265)

[Table de vérité de Parity\_Even\_Out : 4](#_Toc182481266)

[*Equipe 2 : Calcul de la Parité pour les Bits Impairs* 5](#_Toc182481267)

[Table de vérité de Parity\_Odd\_Out 5](#_Toc182481268)

[*Equipe 3 : Combinaison des Parités Calculées* 5](#_Toc182481269)

[Table de vérité de Global\_Parity ? 5](#_Toc182481270)

[*Équipe 4 : Sortie et Indication de la Parité Globale* 6](#_Toc182481271)

[Pour ce dernier fichier, le chef de projet a enregistré les fichiers des 3 premières équipes dans un sous dossier Vivado. Nous avons ensuite combiné l’ensemble dans une top cell. 7](#_Toc182481272)

[*Schématique :* 9](#_Toc182481273)

[*Device :* 10](#_Toc182481274)

[*V.* *GESTION DU PROJET AVEC GITHUB* 10](#_Toc182481275)

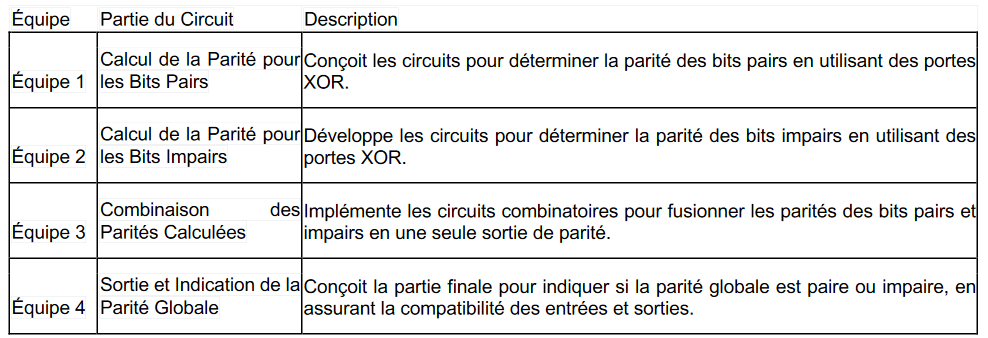
[*VI.* *Conclusion :* 11](#_Toc182481276)

# Description du projet :

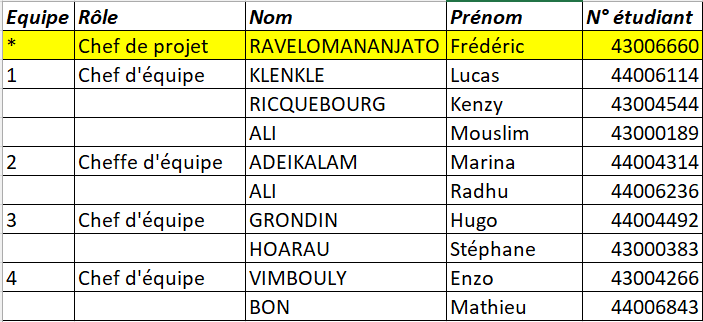
Ce projet mettant en application de circuits combinatoires simples doit nous permettre d’explorer le mode projet et travail d’équipe sous Vivado et GitHub.

Le détecteur de parité vérifie si un nombre binaire a un nombre pair ou impair de bits à 1, produisant une sortie de parité correspondante (nous travaillerons ici avec des mots de 4 digits).

# Détail des différentes parties pour le code VHDL :

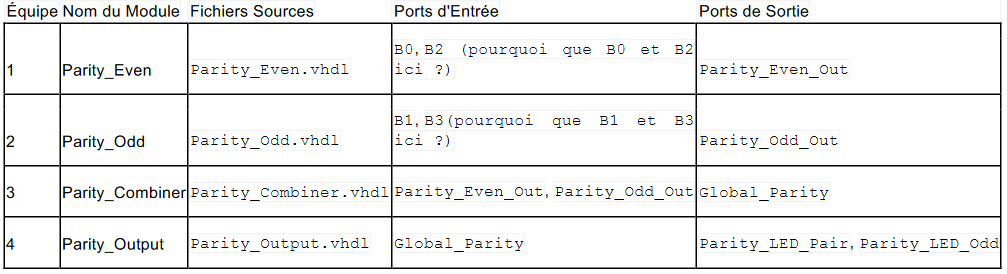


# Présentation des équipes :



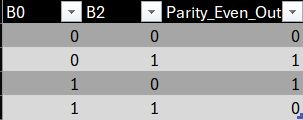
# Application VIVADO

Nous avons donc les consignes pour chaque groupe numéroté respectivement de 1 à 4

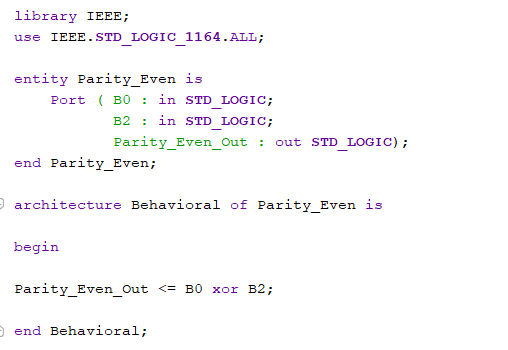


## Equipe 1 : Calcul de la Parité pour les Bits Pairs

### Table de vérité de Parity\_Even\_Out :

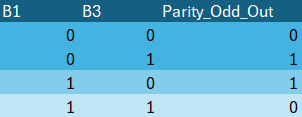


Cette équipe est responsable de la création du module `Parity\_Even`, qui calcule la parité des bits pairs (`B0` et `B2`) en utilisant une porte XOR.

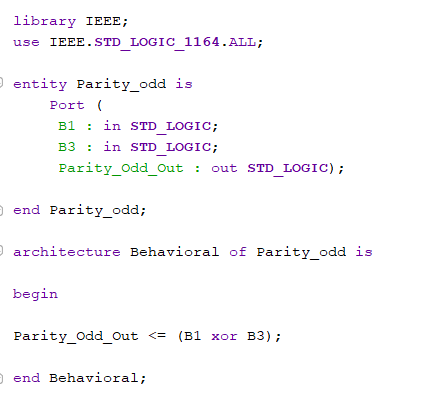


## Equipe 2 : Calcul de la Parité pour les Bits Impairs

### Table de vérité de Parity\_Odd\_Out



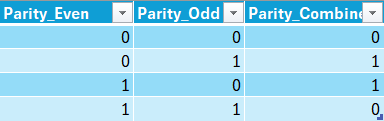
Cette équipe développe le module `Parity\_Odd` qui calcule la parité des bits impairs (`B1` et `B3`) en utilisant une porte XOR.



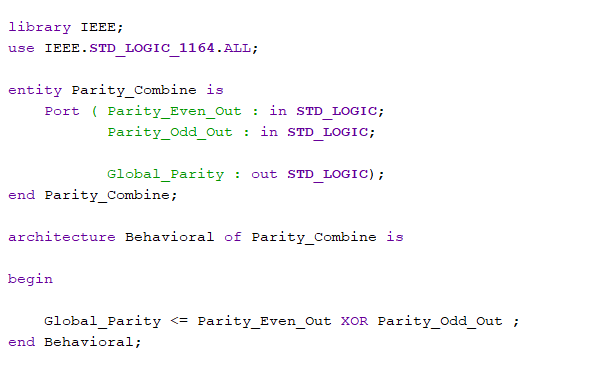
## 

## Equipe 3 : Combinaison des Parités Calculées

### Table de vérité de Global\_Parity ?

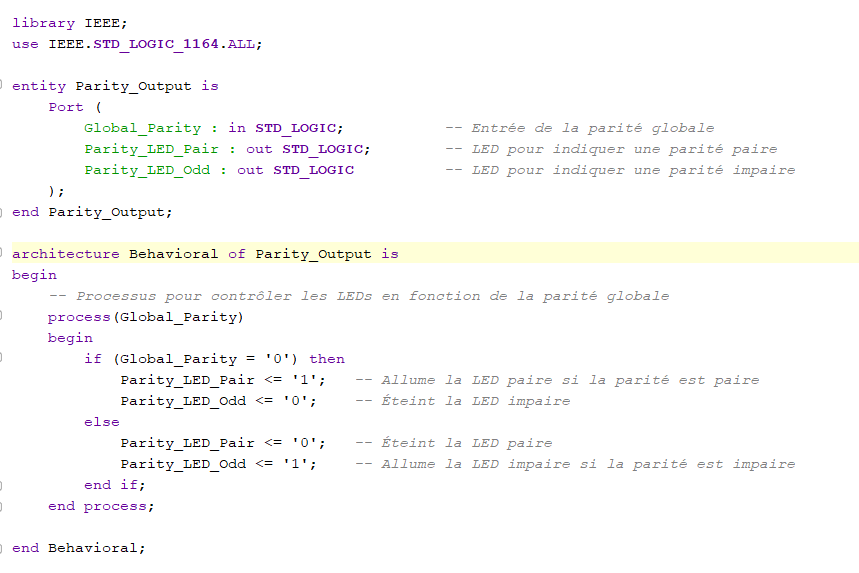


Cette équipe est chargée du module `Parity\_Combiner`, qui combine les parités calculées (`Parity\_Even\_Out` et `Parity\_Odd\_Out`) pour obtenir la parité globale en utilisant une porte XOR.



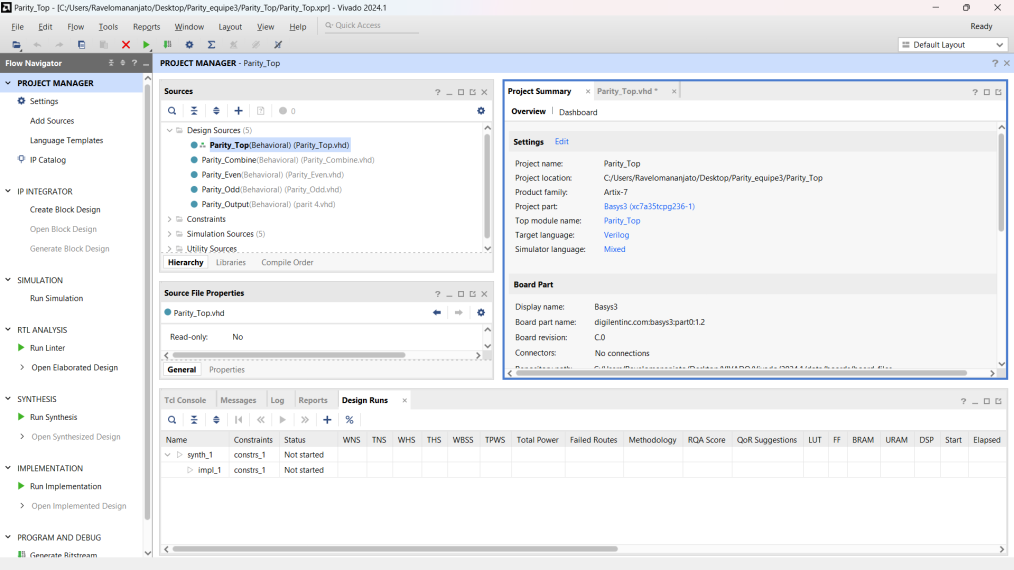
## Équipe 4 : Sortie et Indication de la Parité Globale

Cette équipe développe le module `Parity\_Output`, qui indique visuellement si la parité globale est paire ou impaire en utilisant des LEDs sur la carte Basys 3.



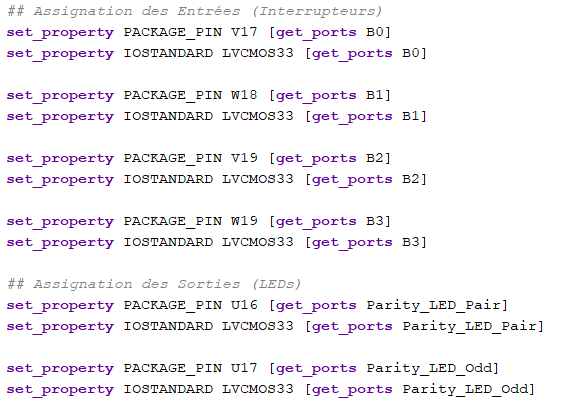
### 

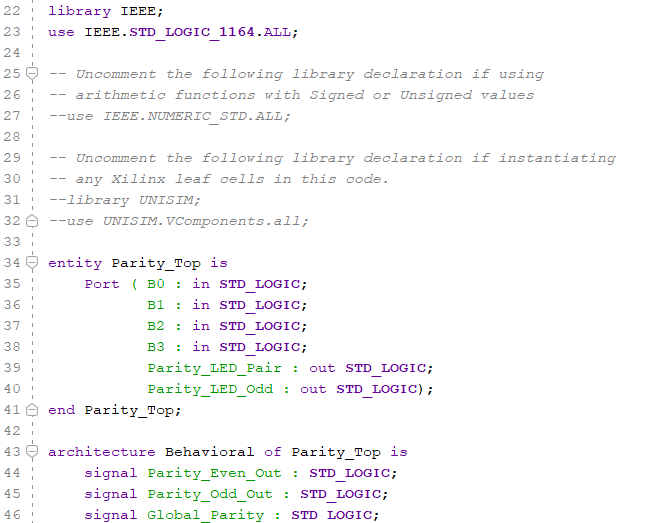
### Pour ce dernier fichier, le chef de projet a enregistré les fichiers des 3 premières équipes dans un sous dossier Vivado. Nous avons ensuite combiné l’ensemble dans une top cell.

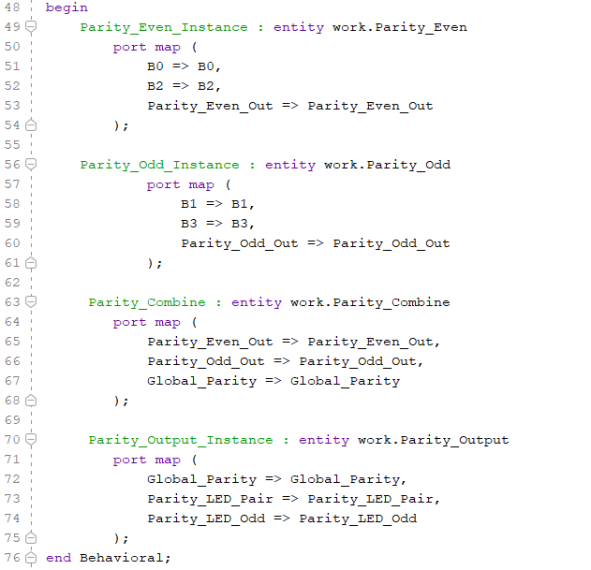


Fichier XDC :

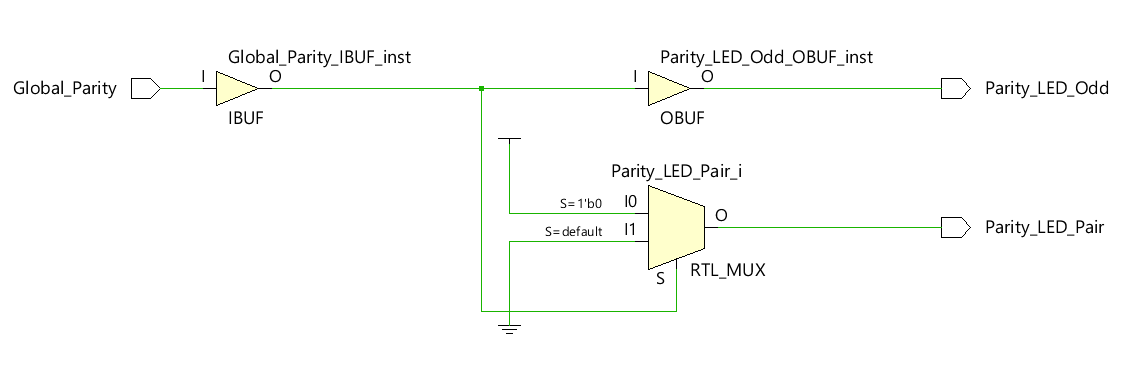
Le chef de projet a assigné les ports aux interrupteurs et leds de la Basys3 et a enregistré le code dans un sous dossier Parity\_Top :





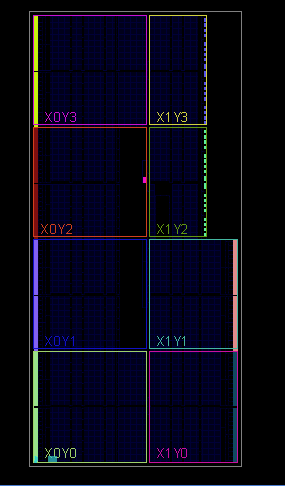


### Schématique :



### 

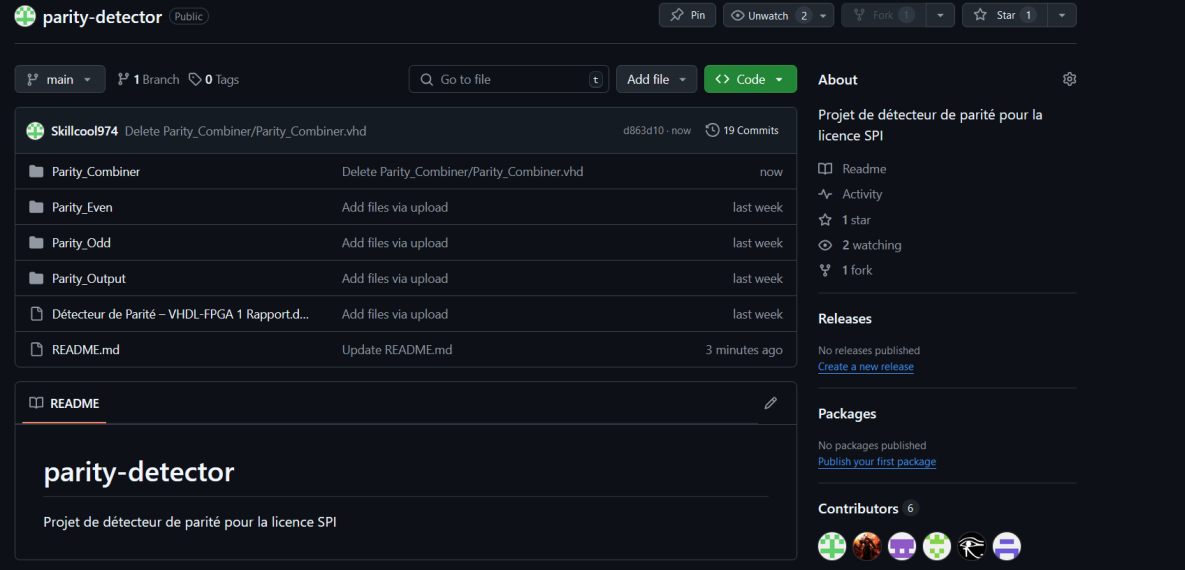
### Device :



# GESTION DU PROJET AVEC GITHUB

Nous avons travaillé sur Github directement sur l’interface sans passer par PowerShell car Github n’était pas encore installé sur nos ordinateurs.

Le chef de projet a créé un dossier sous GitHub. Il a ensuite habilité chaque chef d’équipe. Chaque chef d’équipe y pu ajouter son code dans le sous-dossier attribué. Le chef de projet a ensuite combiné l’ensemble des codes dans une top cell appelé Parity\_Top.



# Conclusion :

Ce projet nous a permis grâce à la réalisation de circuits combinatoires simples de développer nos connaissances en :

* Gestion de projet : organisation et travail en équipes
* Programmation sous Vivado
* Appel à différents codes dans un code combiné
* Gestion de projet via GitHub