

---

# - PLAN DE VALIDATION - ETUDE D'UN SYSTEME COMPLEXE SOC

---

Formation AJC FPGA - Eve CHAR



17 JUILLET 2023  
EVE CHAR

## Table des matières

<b>I.</b>	<b>VUE D'ENSEMBLE DU PROJET .....</b>	<b>2</b>
<b>II.</b>	<b>PLAN DE DEROULEMENT DU PLAN DE VALIDATION .....</b>	<b>3</b>
<b>III.</b>	<b>ENVIRONNEMENT DE TESTS .....</b>	<b>4</b>
<b>IV.</b>	<b>DESCRIPTION DES TESTS .....</b>	<b>4</b>
1.	VALIDATION DE LA FONCTION DE LA LECTURE D'UNE IMAGE.....	4
2.	VALIDATION DE LA FONCTION D'ECRITURE D'UNE IMAGE.....	5
3.	VALIDATION DE LA FONCTION DE GESTION DES PIXELS (FIFO1 ET FIFO2).....	5
a.	Validation des entrées / sorties du fifo1 (lecture- écriture) .....	5
b.	Validation des entrées / sorties du fifo2 (lecture-écriture) .....	7
4.	VALIDATION DE LA FONCTION DE CALCUL DU FILTRE SOBEL.....	8
5.	VALIDATION DE LA FONCTION DE COMPARAISON AVEC UN SEUIL- THRESHOLD.....	9
6.	VALIDATION GLOBALE AVEC LOGICIEL -FIJI .....	9

<i>Création document</i>	<i>Eve CHAR</i>	<i>10 /07/2023</i>
--------------------------	-----------------	--------------------

## I. Vue d'ensemble du projet

Le projet consiste à valider une IP de traitement d'image : Détection de points d'intérêts (corner détection) sur un SOC (système on the Chip).



Détection de points d'intérêt

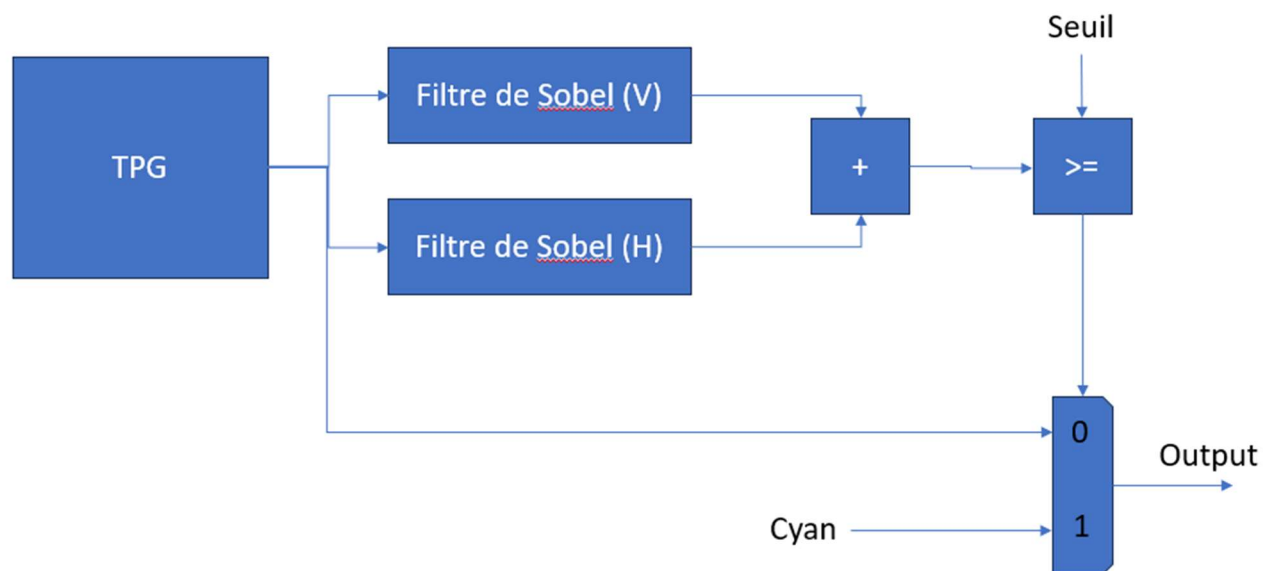


Figure 1: contexte du projet : IP de traitement d'image pour détection de points d'intérêts

## II. Plan de déroulement du plan de validation

Voici un synoptique simplifié de déroulement de plan de validation du projet :

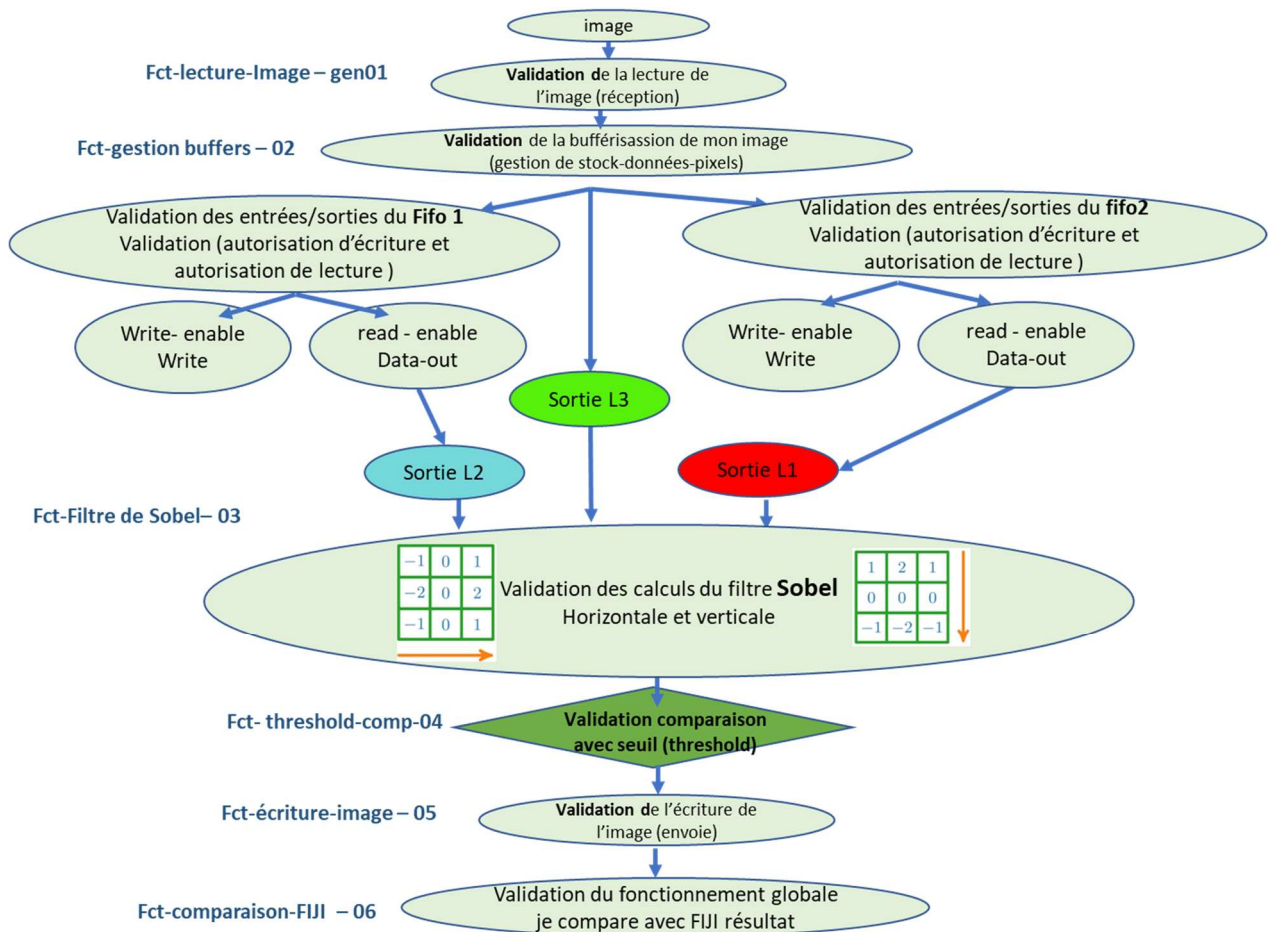


Figure 2 : plan de déroulement du plan de validation

### III. Environnement de tests

Le matériel utilisé lors de la phase de validation est le suivant :


Outil	Utilisation
Test bench xilinx- simulation	Simulation logicielle du code
	Non utilisée
FIJI logiciel	Pour comparer avec les résultats attendus
PC	Calcul de convolution et simulation
Images de tailles différentes	Source à traiter

Figure 3 : Outils à disposition pour réaliser les tests

### IV. Description des tests

#### 1. Validation de la fonction de la lecture d'une image

**Spécification :** image en format « .txt »

**Méthode / comment :** tester une matrice de valeur et voir comment l'outil (testbench) la lit

**Outils :** Logiciel VIVADO-XILINX : Ce logiciel permet de visualiser sous forme de chronogrammes les sorties produites en fonction des entrées forcées, le tout par simulation.

#### Fiches de test

<b>Test N°1 Fct-lecture-Image – gen01 : Vérification de Fct-lecture-Image – gen01</b>		
<i>Résumé: tester une matrice de valeur et voir comment l'outil (tb) la lit</i>		
<i>comment/pré-requis : Réaliser un testbench de la fonction avec :</i>		
<i>en entrée : un fichier de matrice 4 X 4</i>		
<i>et une sortie chronogramme avec les valeur des pixels</i>		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Lancer la simulation	- Valeur des pixels affichés dans le bon ordre de gauche à droite, ligne par ligne

<u>Type d'exécution:</u>	Test_bench
<u>Cahier d'exigences</u>	Fct-lecture-Image – gen01

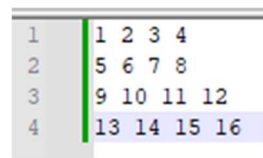
## 2. Validation de la fonction d'écriture d'une image

**Spécification :** écrire données en format « .txt » qui forment une image

**Méthode / comment :** tester l'écriture d'une matrice de valeur et voir comment l'outil (tb) l'écrit

**Outils :** Logiciel VIVADO-XILINX : Ce logiciel permet de visualiser sous forme de chronogrammes les sorties produites en fonction des entrées forcées, le tout par simulation.

### Fiches de test

Test N°2 Fct-écriture-Image – gen05 : Vérification de Fct-écriture-Image – gen05		
<p>Résumé: tester l'écriture d'une matrice de valeur et voir comment l'outil (tb) l'écrit</p> <p>comment/pré-requis : Réaliser un testbench de la fonction avec :</p> <p>en entrée : un fichier de matrice 4 X 4</p> <p>et une sortie chronogramme avec les valeur des pixels et matrice en sortie .txt</p>		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Lancer la simulation	<p>Valeur des pixels affichés dans le bon ordre de gauche à droite, ligne par ligne équivalente à la matrice en entrée : exemple</p> 
<u>Type d'exécution:</u>	Test_bench	
<u>Cahier d'exigences</u>	Fct-écriture-Image – gen05	

## 3. Validation de la fonction de gestion des pixels (Fifo1 et Fifo2)

### a. Validation des entrées / sorties du fifo1 (lecture- écriture)

#### Spécification :

Entrée de la FIFO1 :

- Signal sur 24 bits (input-data)

- Signal d'autorisation d'écriture dans la FIFO1 (write-enable)
- Signal d'autorisation de lecture (rd-enable)

Sortie de la FIFO1 :

- Signal sortie du FIFO1 d-out (pixel\_L2) sur 24 bits

### Méthode / comment :

- Quand la 1e ligne est en train d'être lue :
  - On autorise à écrire dans FIFO1 → (FIFO\_1\_write\_ena)
- Quand la 1e ligne est lue entièrement :
  - On autorise de lire FIFO1 → (FIFO\_1\_read\_ena)

**Outils :** logiciel VIVADO-XILINX et carte CORA Z7

### Fiche de test :

<b>Test N°3</b> test des entrées / sorties du fifo1 (lecture- écriture)		
<p><u>Résumé :</u> Tests permettant de s'assurer de la fonction Fct-gestion buffers – 02</p> <p><u>Comment/pré-requis :</u> Réaliser un test-bench avec :</p> <p>*en entrée : signal <b>input-data</b> et autorisation d'écriture dans le FIFO1</p> <p>*en sortie : FIFO_1_read_ena et Read ( témoin : Init)</p>		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Vérifier l'écriture dans la FIFO1	Regarder quand la 1e ligne est en train d'être lue si (FIFO_1_write_ena) est active
2	Vérifier les conditions de lecture de la fifo : On lit dans FIFO_1 si : - FIFO_1 n'est pas vide - et si la première ligne est entièrement lue	Vérifier quel le signal full est toujours à zéro et que la lecture est autorisée quand Init passe à zéro
3	Vérifier le flux de l'autorisation de lecture de la FIFO1	Regarder quand la première ligne est entièrement reçue : si (FIFO_1_read_ena) est active
<u>Cahier d'exigences</u>	EXIGENCE_ Fct-gestion buffers – 02	



## **b. Validation des entrées / sorties du fifo2 (lecture-écriture)**

### **Spécification :**

Entrée de la FIFO2 :

- Signal (sortie de FIFO1) sur 24 bits
- Signal d'autorisation d'écriture dans la FIFO2 (write-enable)
- Signal d'autorisation de lecture (rd-enable)

Sortie de la FIFO2 :

- Signal sortie du FIFO2 d-out (pixel\_L1) sur 24 bits

### **Méthode / comment :**

- Quand la 1ere ligne est lue entièrement et que la première ligne est entièrement lue :
  - o On autorise à écrire dans FIFO2 → (FIFO\_2\_write\_ena)
- Quand la 2e ligne est lue entièrement :
  - o On autorise de lire FIFO2 → (FIFO\_2\_read\_ena)

**Outils :** logiciel VIVADO-XILINX et carte CORA Z7

### **Fiche de test :**

<b>Test N°4 test des entrées / sorties du fifo2 (lecture- écriture)</b>		
<b>Résumé :</b> Tests permettant de s'assurer de la fonction Fct-gestion buffers – 02		
<b>Comment/pré-requis :</b> Réaliser un test-bench avec :		
<b>*en entrée :</b> signal (sortie de FIFO1) et autorisation d'écriture dans le FIFO2		
<b>*en sortie :</b> signal FIFO_2_read_ena et Read (témoin : Init2)		
<b><u>N° d'étape</u></b>	<b><u>Actions de pas:</u></b>	<b><u>Résultats attendus:</u></b>
1	Vérifier les conditions d'écriture dans la FIFO2	Regarder quand (FIFO_2_write_ena) est active : la 1e ligne est entièrement lue (Init passe à 0) et la ligne 2 en train d'être lue (par FIFO_1)
2	Vérifier les conditions de remplissage de la fifo2 : On écrit dans FIFO2 si :	Vérifier quel le signal full est toujours à zéro Regarder si l'autorisation d'écriture passe à 1



	- FIFO2 n'est pas rempli - et si la première est entièrement lue	
3	Vérifier le flux de l'autorisation de lecture de la FIFO2	FIFO_2_read_ena est active quand la deuxième ligne est entièrement lue Regarder si l'autorisation de lecture passe à 1
<u>Cahier d'exigences</u>	EXIGENCE_ Fct-gestion buffers – 02	

#### 4. Validation de la fonction de calcul du Filtre Sobel

**Spécification :** Il faut multiplier les coefficients du masque par les valeurs des pixels, réaliser la somme puis comparer au Threshold. La comparaison au threshold sera dans un autre test, concentrons-nous ici à vérifier le calcul de la somme.

**Méthode / comment :**

**Outils :** logiciel VIVADO-XILINX

**Fiche de test :**

Test N°5 fonction : Fct- horizontale-Filtre de Sobel– 03		
<u>Résumé:</u> Tests permettant de s'assurer que la fonction Filtre de Sobel– 03 se réalise comme attendu. <u>comment/pré-requis :</u> Valeurs des pixels issus du fichier. Visualiser la variable de sortie du module. <u>*en entrée :</u> Valeurs des pixels issus du fichier <u>*en sortie :</u> variable de sortie du module de convolution		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Lancer la simulation	Vérifier le calcul
<u>Cahier d'exigences</u>	Fct- horizontale-Filtre de Sobel– 03	

## 5. Validation de la fonction de comparaison avec un seuil- threshold

**Spécification :** la sortie du filtre de Sobel est comparée à une valeur seuil (threshold), on peut alors déterminer les points de changement soudain de luminosité, qui correspond à des bords/ contours.

**Méthode / comment :** on compare la sortie de filtre avec une valeur seuil (intensité). En fonction du résultat, on sélectionne une valeur de pixel « cyan » si supérieur ou égale au seuil sinon on prend la valeur du pixel d'origine.

**Outils :** Logiciel VIVADO-XILINX : Ce logiciel permet de visualiser sous forme de chronogrammes les sorties produites en fonction des entrées forcées, le tout par simulation.

### Fiche de test :

Test N°6 fonction : Fct- threshold-comp-04		
<p><u>Résumé :</u> Tests permettant de vérifier la comparaison : si la sortie du filtre de Sobel est <math>\geq</math> threshold, alors on prend une valeur de pixel « cyan » sinon on prend la valeur du pixel d'origine.</p> <p><u>Comment/pré-requis :</u> Lancer la simulation</p> <p>*en entrée : image_in.txt, Threshold, sortie du filtre</p> <p>*en sortie : image_out.txt, output_data</p>		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Vérifier la comparaison avec threshold	Si la sortie du filtre de Sobel est $\geq$ threshold, alors on prend une valeur de pixel « cyan » sinon on prend la valeur du pixel d'origine.
<u>Type d'exécution :</u>	Test_bench	
<u>Cahier d'exigences</u>	EXIGENCE_ Fct- threshold-comp-04	

## 6. Validation globale avec logiciel -FIJI

**Spécification :** le logiciel FIJI permet d'anticiper les résultats à obtenir après application du filtre SOBEL.

**Méthode / comment :** comparaison de résultat obtenu avec notre test Bench et celles obtenue avec le logiciel FIJI.

**Outils :** logiciel VIVADO-XILINX : Ce logiciel permet de visualiser sous forme de chronogrammes les sorties produites en fonction des entrées forcées, le tout par simulation.

**Fiche de test :**

<b>Test N°7 fonction : Fct-comparaison globale FIJI – 06</b>		
<p><u>Résumé :</u> Tests permettant la comparaison de résultat obtenu avec notre test Bench et celles obtenue avec le logiciel FIJI.</p> <p><u>Comment/pré-requis :</u> Réaliser un test-bench de la fonction <b>Fct-comparaison globale FIJI – 06</b> avec :</p> <p>*en entrée : image_in.txt, Threshold, sortie du filtre</p> <p>*en sortie : image_out.txt, output_data</p>		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Comparer les images de test bench avec celle de FIJI	Les mêmes points de contours marqués
<u>Type d'exécution :</u>	Test_bench	
<u>Cahier d'exigences</u>	EXIGENCE_ Fct-comparaison globale FIJI – 06	