

- PLAN DE VALIDATION- SEGMENT 4 : ETUDE D'UN SYSTEME COMPLEXE SOC

Formation AJC FPGA - Eve CHAR



XX JUILLET 2023
EVE CHAR

Table des matières

I.	VUE D'ENSEMBLE DU PROJET	2
II.	PLAN DE DEROULEMENT DU PLAN DE VALIDATION	3
III.	ENVIRONNEMENT DE TESTS	4
IV.	DESCRIPTION DES TESTS	4
1.	VALIDATION FCT-LECTURE-IMAGE – GEN01	4
2.	VALIDATION DE LA FCT-GESTION BUFFERS – 02 (FIFO1 ET FIFO2)	5
a.	<i>Validation des entrées / sorties du fifo1 (lecture- ecriture)</i>	<i>5</i>
b.	<i>Validation des entrées / sorties du fifo2 (lecture-écriture)</i>	<i>6</i>
3.	VALIDATION FCT- HORIZONTALE-FILTRE DE SOBEL– 03.....	8
4.	VALIDATION FCT- VERTICAL-FILTRE DE SOBEL– 04.....	8
5.	VALIDATION DE LA FCT- THRESHOLD-COMP-05.....	9
6.	VALIDATION DE LA FCT-ECRITURE-IMAGE – 06	10
7.	VALIDATION DE LA FCT-COMPARAISON-FIJI – 07	11

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

I. Vue d'ensemble du projet

Le projet consiste à valider d'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

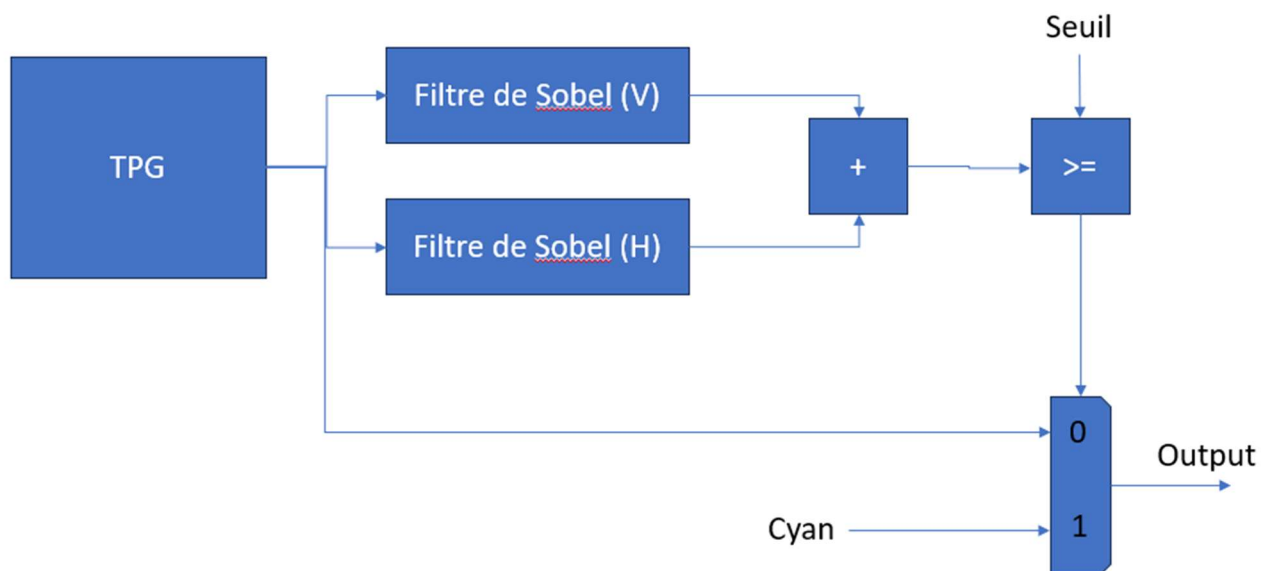


Figure 1: contexte du projet : IP de traitement d'image pour détection de points d'intérêts

II. Plan de déroulement du plan de validation

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

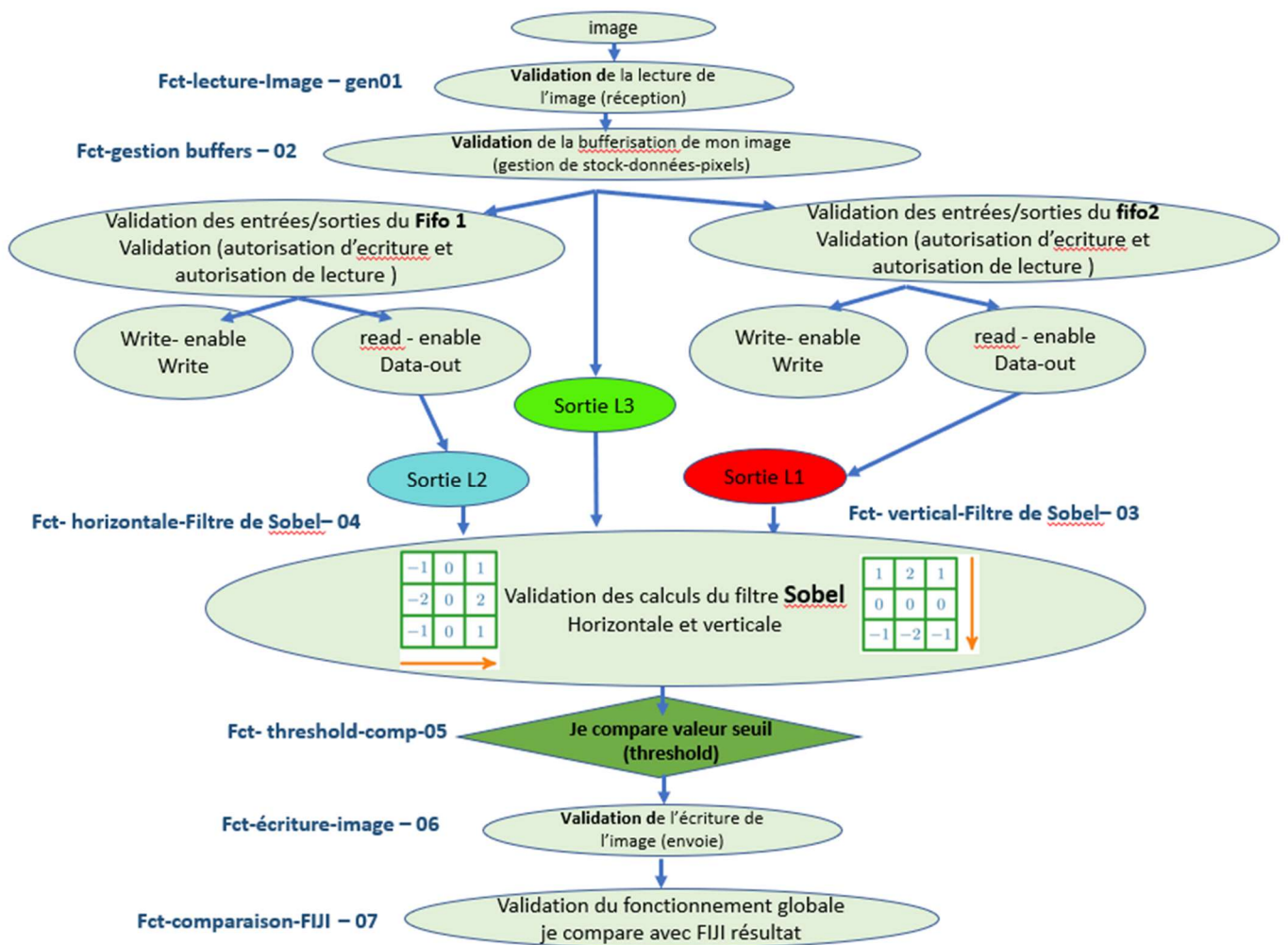


Figure 2 : plan de déroulement du plan de validation

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 Fct-lecture-Image – gen01

Spécification :

Méthode / comment :

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°1 Fct-lecture-Image – gen01 : Vérification de Fct-lecture-Image – gen01		
<u>Résumé:</u> Tests permettant de s'assurer que		
<u>comment/pré-requis :</u> Réaliser un testbench de la fonction avec en entrée xxx et une sortie xxx		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Lancer la simulation	

<u>Type d'exécution:</u>	Test_bench
<u>Durée estimée d'exéc. (en min):</u>	30 min
<u>Cahier d'exigences</u>	Fct-lecture-Image – gen01

2. Validation de la Fct-gestion buffers – 02 (Fifo1 et Fifo2)

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

Spécification :

Entrée de la FIFO1 :

- Signal (pixel_L3) sur 24 bits (sortie de XX)
- Signal d'autorisation d'écriture dans la FIFO1 (write-enable)
- Signal d'autorisation de lecture (rd-enable)

Sortie de la FIFO1 :

- Signal sortie de 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 :

Test N°9 test des entrées / sorties du fifo1 (lecture- écriture)
<u>Résumé :</u> Tests permettant de s'assurer que la fonction Fct-gestion buffers – 02
<u>Comment/pré-requis :</u> Réaliser un test-bench avec :
*en entrée : signal sortie (24 bits) et autorisation d'écriture dans le FIFO1
*en sortie : signal data RGB : couleur sur (24 bits)

<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Vérifier la valeur du pixel_L3 (l'écriture dans la FIFO)	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 (Init doit être à 0)	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 lue : si (FIFO_1_read_ena) est active (h_cnt = xx)
<u>Type d'exécution :</u>	Test_bench	
<u>Durée estimée d'exéc (en min) :</u>	60 min	
<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 (pixel-L3) sur 24 bits (sortie de FIFO1)
- Signal d'autorisation d'écriture dans la FIFO2 (write-enable)
- Signal d'autorisation de lecture (rd-enable)

Sortie de la FIFO2 :

- Signal sortie de d-out (pixel-L2) sur 24 bits

Méthode / comment :

- Quand la 1ere ligne est lue entièrement et que la première ligne est entièrement lue :

- On autorise à écrire dans FIFO2 → (FIFO_2_write_ena)
- Quand la 2e ligne est lue entièrement :
 - On autorise de lire FIFO2 → (FIFO_2_read_ena)

Outils : logiciel VIVADO-XILINX et carte CORA Z7

Fiche de test :

Test N°10 test des entrées / sorties du fifo2 (lecture- écriture)		
<u>Résumé</u> : Tests permettant de s’assurer que la fonction Fct-gestion buffers – 02		
<u>Comment/pré-requis</u> : Réaliser un test-bench de la fonction Fct-gestion buffers – 02 avec :		
*en entrée : signal sortie de xx et autorisation d’écriture dans le FIFO1		
*en sortie : signal data RGB : couleur sur (24 bits)		
N° d’étape	Actions de pas:	Résultats attendus:
1	Vérifier les conditions d’écriture dans la FIFO_2	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 : - FIFO2 n'est pas rempli - et si la première est entièrement lue	Vérifier quel le signal full est toujours à zéro Regarder si l’autorisation d’écriture passe à 1 pour v_cnt = 0 et h-cnt = xxx
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 pour v_cnt = 1 et h-cnt = xxx
<u>Type d’exécution</u> : Test_bench		
<u>Durée estimée d'exéc (en min)</u> : 60 min		
<u>Cahier d’exigences</u>	EXIGENCE_ Fct-gestion buffers – 02	

3. Validation Fct- horizontale-Filtre de Sobel– 03

Spécification :

Méthode / comment :

Outils : logiciel VIVADO-XILINX

Fiche de test :

Test N°3 fonction : Fct- horizontale-Filtre de Sobel– 03		
<u>Résumé:</u> Tests permettant de s’assurer que la fonction xx		
Quoi :		
comment/pré-requis :		
*en entrée :		
*en sortie :		
- Lancer la simulation		
<u>N° d’étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
<u>Type d’exécution:</u>	Test_bench	
<u>Durée estimée d'exéc. (en min):</u>	60 min	
<u>Cahier d’exigences</u>	Fct- horizontale-Filtre de Sobel– 03	

4. Validation Fct- vertical-Filtre de Sobel– 04

Spécification :

Méthode / comment :

Outils : logiciel VIVADO-XILINX

Fiche de test :

Test N°3 fonction : Fct- vertical -Filtre de Sobel– 04		
--	--	--

Résumé: Tests permettant de s'assurer que la fonction xx

Quoi :

comment/pré-requis :

*en entrée :

*en sortie :

- Lancer la simulation

<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
<u>Type d'exécution:</u>	Test_bench	
<u>Durée estimée</u> <u>d'exéc. (en min):</u>	60 min	
<u>Cahier d'exigences</u>	Fct- vertical -Filtre de Sobel– 04	

5. Validation de la Fct- threshold-comp-05

Spécification :

Méthode / comment :

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°5 fonction : Fct- threshold-comp-05: Vérification de la comparaison de xxxx		
<u>Résumé :</u> Tests permettant de s'assurer que la fonction <u>Comment/pré-requis :</u> Réaliser un test-bench de la fonction <u>*en entrée :</u> <u>*en sortie :</u>		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>

1		
Type d'exécution :	Test_bench	
Durée estimée d'exéc (en min) :	60 min	
Cahier d'exigences	EXIGENCE_ compteur_H_POS_05	

6. Validation de la Fct-écriture-image – 06

Spécification :

Méthode / comment :

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-écriture-image – 06: Vérification de la position		
<u>Résumé :</u> Tests permettant de s'assurer que la fonction compteur_V_POS_06 compte correctement la position pixel sur la ligne horizontale <u>Comment/pré-requis :</u> Réaliser un test-bench de la fonction compteur_V_POS_06 avec : *en entrée : pixel_clock et RESET *en sortie : signaux logiques de synchronisation (v_sync)		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1		
2		
<u>Type d'exécution :</u>	Test_bench	
<u>Durée estimée d'exéc (en min) :</u>	60 min	

<u>Cahier d'exigences</u>	EXIGENCE_compteur_V_POS_06

7. Validation de la Fct-comparaison-FIJI – 07

Spécification :

Méthode / comment :

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°7 fonction : display_area_07 : Vérification de la zone visible de l'écran		
<u>Résumé</u> : Tests permettant de s'assurer que la fonction display_area_07 affiche la bonne zone visible		
<u>Comment/pré-requis</u> : Réaliser un test-bench de la fonction display_area_07 avec :		
*en entrée : pixel_clock et RST		
*en sortie : signaux logiques de synchronisation (h_sync et v_sync avec leurs compteurs respectives)		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Vérifier la largeur de la zone visible	
2		
<u>Type d'exécution</u> :	Test_bench	
<u>Durée estimée d'exéc (en min)</u> :	20 min	
<u>Cahier d'exigences</u>	EXIGENCE_display_area_07	