

- CAHIER DE RECETTE - ETUDE D'UN SYSTEME COMPLEXE SOC

Formation AJC FPGA - Eve CHAR



17 JUILLET 2023
EVE CHAR

Table des matières

I.	CRITERES D'ACCEPTATION DES ESSAIS.....	2
II.	RESULTATS DE TESTS	2
1.	VALIDATION DE LA FONCTION DE LA LECTURE D'UNE IMAGE.....	2
2.	VALIDATION DE LA FONCTION D'ECRITURE D'UNE IMAGE.....	3
3.	VALIDATION DE LA FONCTION DE GESTION DES PIXELS (FIFO1 ET FIFO2).....	5
3.1	Validation des entrées / sorties du fifo1 (lecture- écriture)	5
3.2	Validation des entrées / sorties du fifo2 (lecture- ecriture)	6
4.	VALIDATION DE LA FONCTION DE CALCUL DU FILTRE SOBEL.....	7
5.	VALIDATION DE LA FONCTION DE COMPARAISON AVEC UN SEUIL- THRESHOLD.....	8
6.	VALIDATION GLOBALE AVEC LOGICIEL -FIJI	9

Création document	Eve CHAR	11 /07/2023
-------------------	----------	-------------

I. Critères d'acceptation des essais

Pour chaque essai, l'activation et l'acceptation est basée sur les critères suivants :

- ✓ Le test est activable :
 - 1) les données en entrée sont prêtes,
 - 2) les outils, fonctions, systèmes nécessaires sont disponibles,
 - 3) la procédure d'essai peut être déroulée,
 - 4) les essais précédents dont dépend cet essai se sont déroulés avec succès.
- ✓ Les sorties de l'essai sont conformes aux sorties spécifiées.

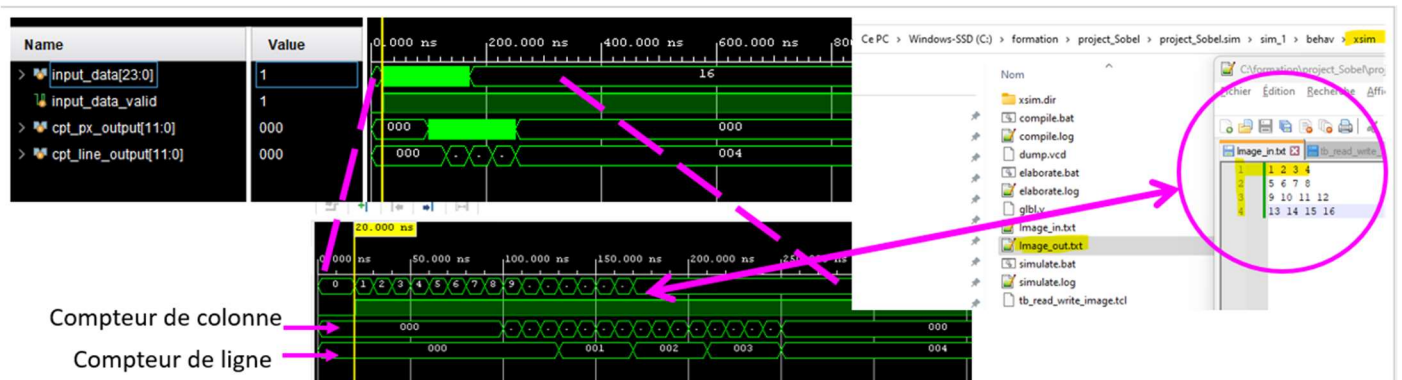
Le test est refusé si une des conditions ci-dessus n'est pas respectée.

II. Résultats de tests

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

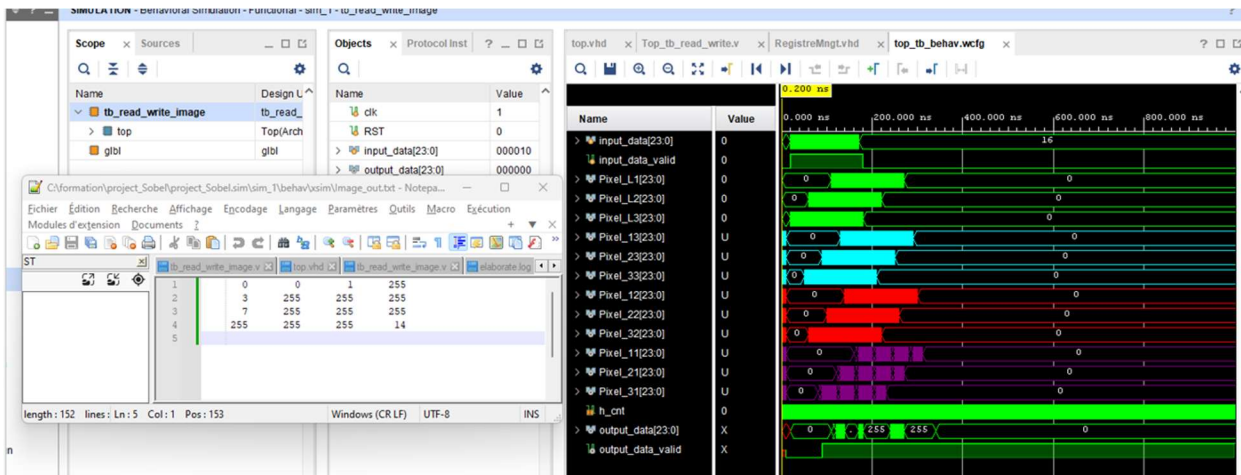
Résultats du test :

Test N°1 Fct-lecture-Image – gen01 : Vérification de Fct-lecture-Image – gen01			
<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>	<u>Résultats obtenus :</u>
1	Lancer la simulation	Valeur des pixels affichés dans le bon ordre de gauche à droite ligne par ligne	Ok 
<u>Type d'exécution:</u>	Test_bench		
<u>Durée estimée d'exéc. (en min):</u>	30 min		
			✓ Pass
<u>Cahier d'exigences</u>	Fct-lecture-Image – gen01		

Démonstration :**2. Validation de la fonction d'écriture d'une image****Résultats du test :**

Test N°2 Fct-écriture-Image – gen05 : Vérification de Fct-écriture-Image – gen05			
Résumé: tester l'écriture d'une matrice de valeur et voir comment l'outil (tb) l'écrit			
comment/pré-requis : Réaliser un testbench de la fonction avec :			
en entrée : un fichier de matrice 4 X 4			
et une sortie chronogramme avec les valeur des pixels et matrice en sortie .txt			
N° d'étape	Actions de pas:	Résultats attendus:	Résultats obtenus :
1	Lancer la simulation	Valeur des pixels affichés dans le bon ordre de gauche à droite, ligne par ligne équivalente à la matrice en entrée <pre> 1 1 2 3 4 2 5 6 7 8 3 9 10 11 12 4 13 14 15 16 </pre>	<div> <div>Ecriture :image -out</div> <div> <pre> 1 1 2 3 4 2 5 6 7 8 3 9 10 11 12 4 13 14 15 16 </pre> </div> </div> <div> <div>lecture :image -in</div> <div> <pre> 1 1 2 3 4 2 5 6 7 8 3 9 10 11 12 4 13 14 15 16 </pre> </div> </div> <p>Ok</p>
Type d'exécution:	Test_bench		
Cahier d'exigences	Fct-écriture-Image – gen05		✓ Pass

Démonstration :



Ce PC > Windows-SSD (C:) > formation > project_Sobel > project_Sobel.sim > sim_1 > behav > xsim

Nom	Modifié le	Type	Taille
xsim.dir	11/07/2023 11:50	Dossier de fichiers	
compile.bat	11/07/2023 16:26	Fichier de comman...	2 Ko
compile.log	11/07/2023 16:26	Fichier LOG	1 Ko
dump.vcd	11/07/2023 16:26	Fichier VCD	0 Ko
elaborate.bat	11/07/2023 16:26	Fichier de comman...	2 Ko
elaborate.log	11/07/2023 16:26	Fichier LOG	2 Ko
glib.v	18/11/2020 05:31	Fichier V	2 Ko
Image_in.txt	11/07/2023 12:18	Fichier TXT	1 Ko
Image_out.txt	11/07/2023 16:26	Fichier TXT	1 Ko

Ecriture :image -out

1	1	2	3	4
2	5	6	7	8
3	9	10	11	12
4	13	14	15	16
5				

lecture :image -in

1	1	2	3	4
2	5	6	7	8
3	9	10	11	12
4	13	14	15	16



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

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

Résultats du test :

Test N°3 test des entrées / sorties du fifo1 (lecture- écriture)			
<u>Résumé</u> : Tests permettant de s'assurer de la fonction Fct-gestion buffers – 02 <u>Comment/pré-requis</u> : Réaliser un test-bench avec : *en entrée : signal input-data et autorisation d'écriture dans le FIFO1 *en sortie : FIFO_1_read_ena et Read (témoin : Init)			
N° d'étape	Actions de pas :	Résultats attendus :	Résultat obtenu
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	OK
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	OK
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	Ok
<u>Type d'exécution</u> :	Test_bench		
<u>Cahier d'exigences</u>	EXIGENCE_ Fct-gestion buffers – 02		✓ Pass

Démonstration :

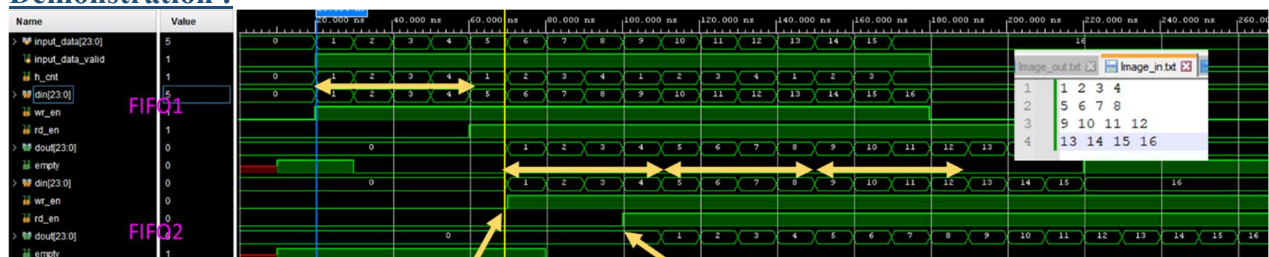


3.2 Validation des entrées / sorties du fifo2 (lecture- écriture)

Résultats du test :

Test N°4 test des entrées / sorties du fifo2 (lecture- écriture)			
<u>Résumé :</u> Tests permettant de s'assurer de la fonction Fct-gestion buffers – 02 <u>Comment/pré-requis :</u> Réaliser un test-bench avec : *en entrée : signal (sortie de FIFO1) et autorisation d'écriture dans le FIFO2 *en sortie : signal FIFO_2_read_ena et Read (témoin : Init2)			
N° d'étape	Actions de pas:	Résultats attendus:	Résultat obtenu
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)	OK
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	OK
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	Ok
<u>Cahier d'exigences</u> EXIGENCE_ Fct-gestion buffers – 02			
			✓ Pass

Démonstration :



Ecrire dans FIFO 2 Fin de la ligne1

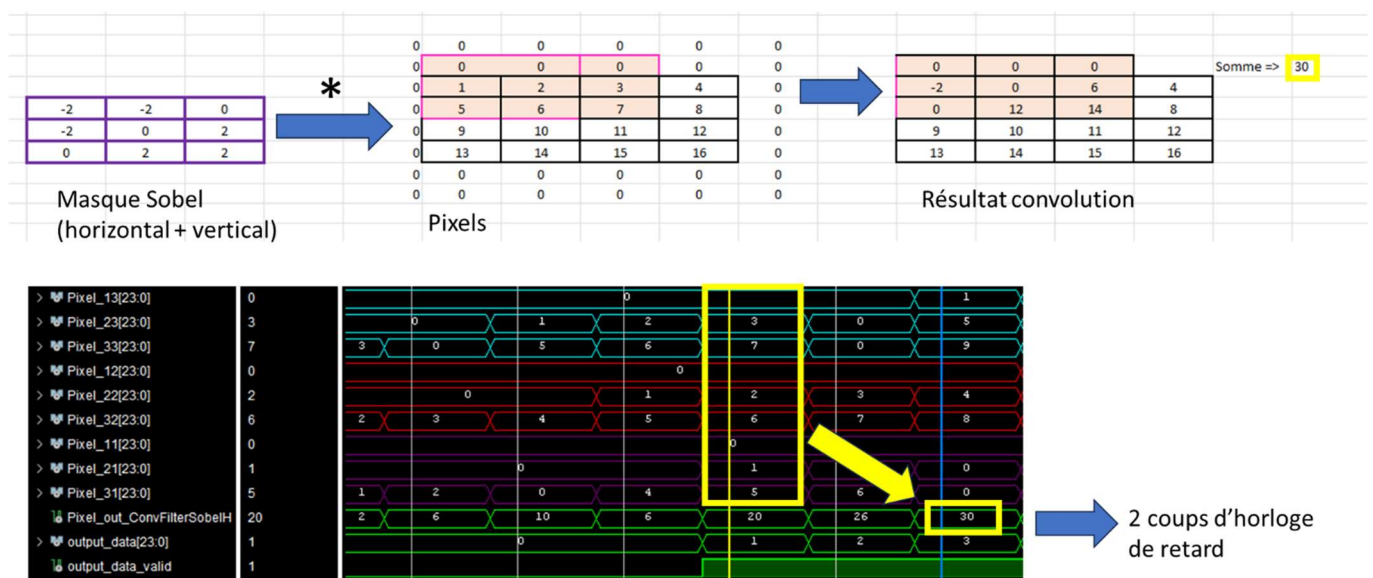
Lire de la FIFO 2 Fin de la ligne2

4. Validation de la fonction de calcul du Filtre Sobel

Résultats du test :

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

Démonstration :



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

Résultats du 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 \geq 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>			
N° d'étape	Actions de pas :	Résultats attendus :	Résultat obtenu
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.	Ok
<u>Cahier d'exigences</u>	EXIGENCE_ Fct- threshold-comp-04		✓ Pass

Démonstration :

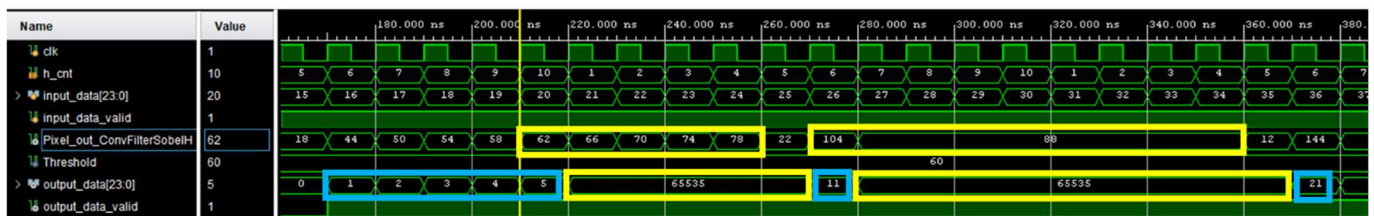
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40

Image_in.txt

```
constant Threshold : integer := 60;
constant Pixel_Cyan : std_logic_vector(23 downto 0) := ("000000001111111111111111");
```

1	2	3	4	5	65535	65535	65535	65535	65535
11	65535	65535	65535	65535	65535	65535	65535	65535	65535
21	65535	65535	65535	65535	65535	65535	65535	65535	65535
65535	32	65535	65535	65535	65535	65535	65535	65535	65535

Image_out.txt

cyan = 65535 = (2¹⁶)-1

6. Validation globale avec logiciel -FIJI

Résultats du test :

Test N°7 fonction : Fct-comparison globale FIJI – 06			
<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 Fct-comparison globale FIJI – 06 avec :</p> <p>*en entrée : image_in.txt, Threshold, sortie du filtre</p> <p>*en sortie : image_out.txt, output_data</p>			
N° d'étape	Actions de pas:	Résultats attendus:	Résultat obtenu
1	Comparer les images de test bench avec celle de FIJI	Les mêmes points de contours marqués	OK
<u>Cahier d'exigences</u>	EXIGENCE_ Fct-comparison globale FIJI – 06	✓	PASS

Démonstration :

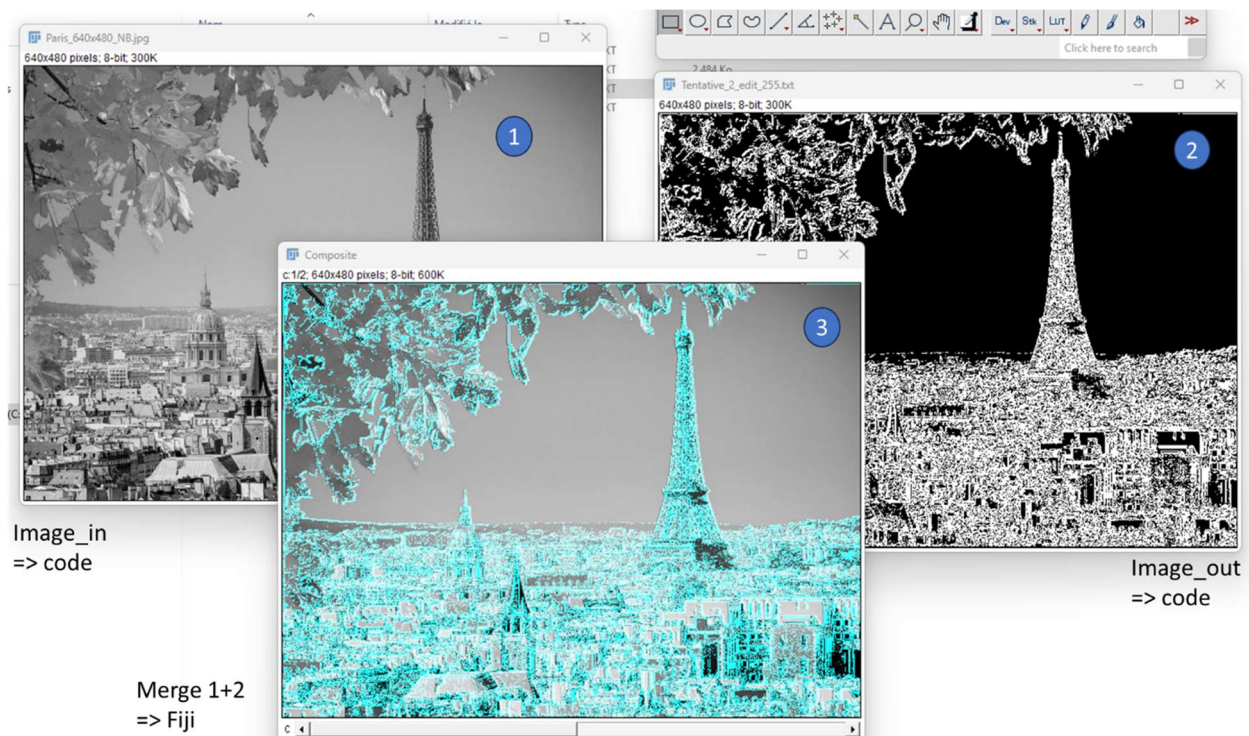
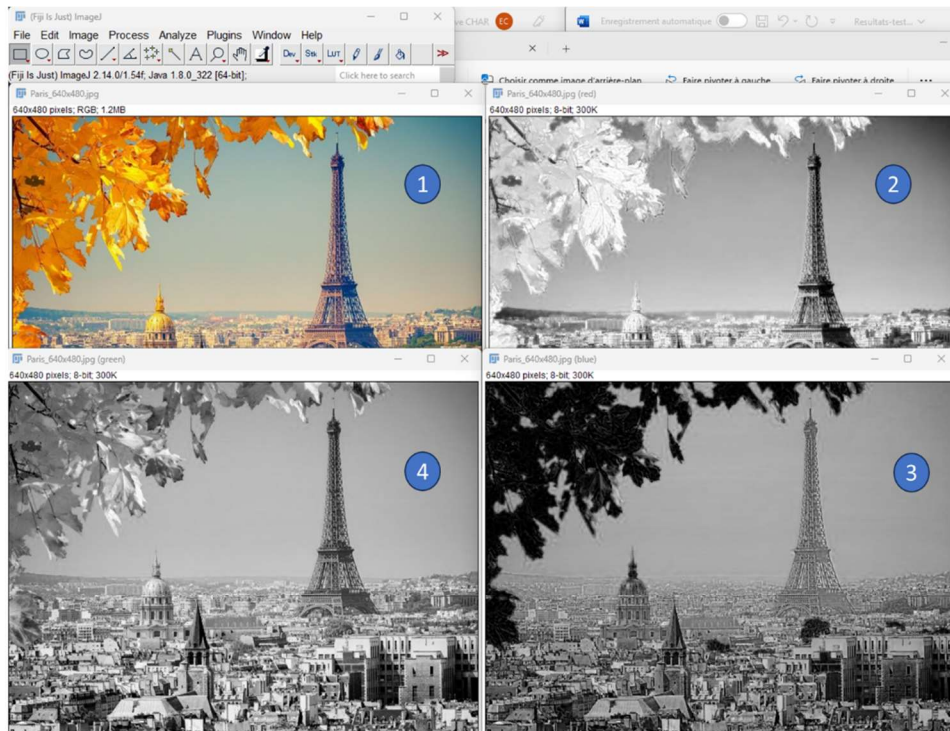
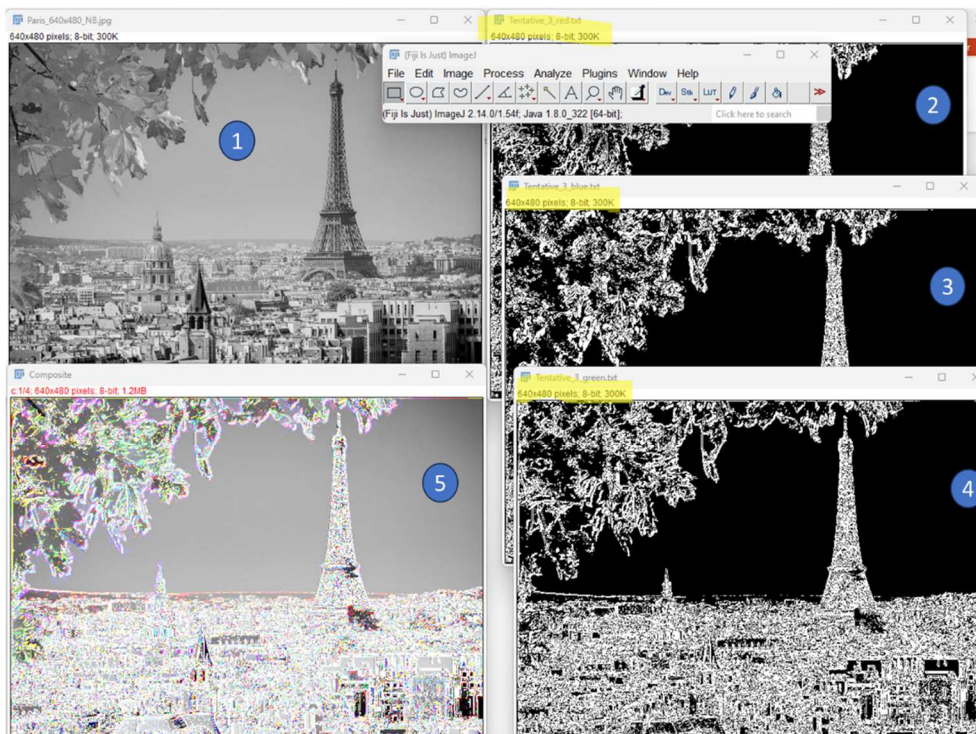


Image
d'origine
splittée selon
les 3 plans de
couleur RGB



Images
d'entrée du
code sur les
plans
2 – red
3 – blue
4 – green

Image
d'origine en
nuance de
gris



Images de
sortie du code
sur les plans
2 – red
3 – blue
4 – green

Merge des
contours
selon les 3
plans (2 / 3
/ 4) sur
l'image
d'origine en
nuance de
gris (1)