



-PLAN DE VALIDATION- INTERFACE VIDEO VGA

Formation AJC FPGA - Eve CHAR



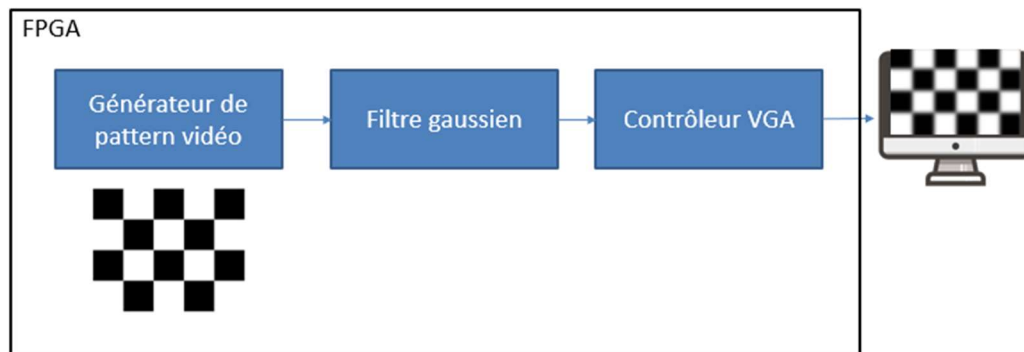
03 JUILLET 2023
EVE CHAR

Table des matières

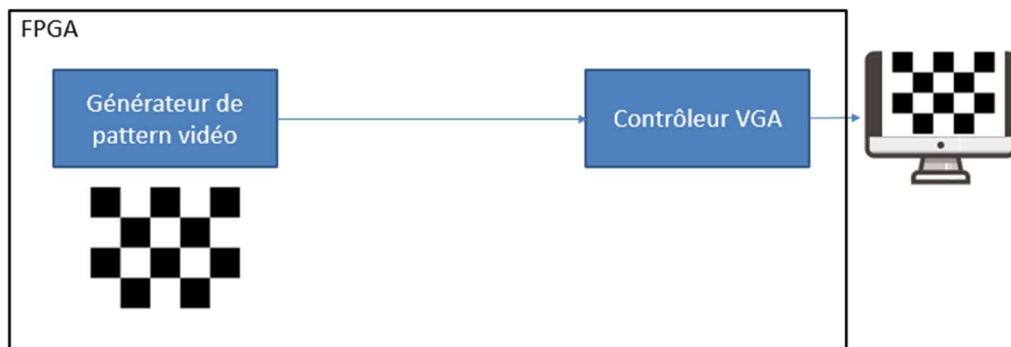
I.	VUE D'ENSEMBLE DU PROJET	2
II.	PLAN DE DEROULEMENT DU PLAN DE VALIDATION	3
1.	PHASE 1 : ETAPE INTERMEDIAIRE (SANS FILTRE)	3
2.	PHASE 2 : AJOUT D'UN FILTRE FIR (GAUSS)	4
III.	ENVIRONNEMENT DE TESTS	5
IV.	DESCRIPTION DES TESTS PHASE 1 : CONTROLEUR VGA SANS FILTRE DE GAUSS	7
1.	VALIDATION DE LA GENERATION DE L'HORLOGE VGA « PIXEL CLOCK ».....	7
2.	VALIDATION DE LA SYNCHRONISATION HORIZONTALE (H-SYNC)	8
3.	VALIDATION DE LA SYNCHRONISATION VERTICALE (V-SYNC)	10
4.	VALIDATION DE L'ENSEMBLE HSYNC ET VSYNC : TIMING GENERALE (TIMING_GEN).....	13
5.	VALIDATION DE LA POSITION DU PIXEL (H_CNT) ET SON RESET.....	15
6.	VALIDATION DE LA POSITION DU PIXEL SUR UNE TRAME (V_CNT) ET SON RESET.....	17
7.	VALIDATION DE LA ZONE D'AFFICHAGE DE L'ECRAN (VIDEO_ON).....	18
8.	VALIDATION GLOBALE DE LA PHASE 1 « CONTROLEUR_VGA » SANS FILTRE	20
V.	DESCRIPTION DES TESTS - PHASE 2 : CONTROLEUR VGA AVEC FILTRE DE GAUSS.....	20
1.	VALIDATION DES ENTREES / SORTIES DU FIFO1 (LECTURE- ECRITURE).....	20
2.	VALIDATION DES ENTREES / SORTIES DU FIFO2 (LECTURE-ECRITURE).....	22
3.	VALIDATION DES ENTREES / SORTIES DU CONV-FILTER-GAUSS (CALCUL).....	23
4.	VALIDATION DE LA SYNCHRONISATION DU SYSTEME APRES L'AJOUT DU FILTRE.....	25
5.	VALIDATION DU DEGRADE DE GRIS DANS LES BORDS APRES L'AJOUT DU FILTRE	26
6.	VALIDATION GLOBALE DE LA PHASE 2 « CONTROLEUR_VGA » AVEC FILTRE	28
VI.	MOYEN DE TEST 2 :TEST SUR CIBLE -SET UP -DEBUG - ILA	28
VII.	MOYEN DE TEST 3 : TEST SUR CIBLE - OSCILLOSCOPE	32

I. Vue d'ensemble du projet

Le projet consiste à faire la validation d'une IP de traitement d'image et d'affichage VGA sur cible Zynq7010 :



Etape intermédiaire :



Phase finale :



Figure 1: contexte du projet : IP de traitement d'image sur cible Zynq7010 et affichage VGA

II. Plan de déroulement du plan de validation

1. Phase 1 : étape intermédiaire (sans filtre)

Voici un synoptique simplifié de déroulement de plan de validation de la première partie du projet :

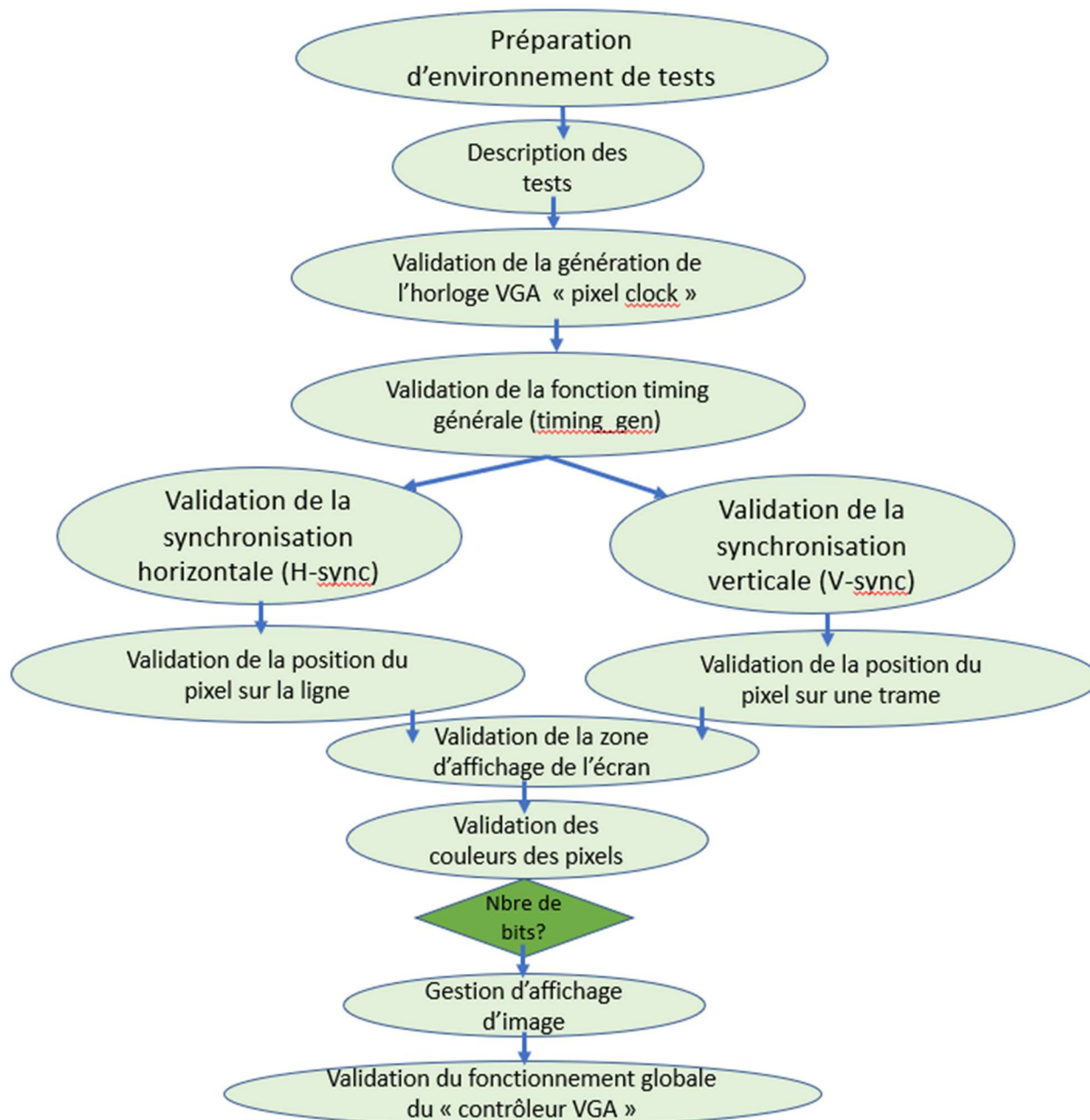


Figure 2 : plan de déroulement du plan de validation

2. Phase 2 : ajout d'un filtre FIR (Gauss)

Voici un synoptique simplifié de déroulement de plan de validation de la deuxième partie du projet :

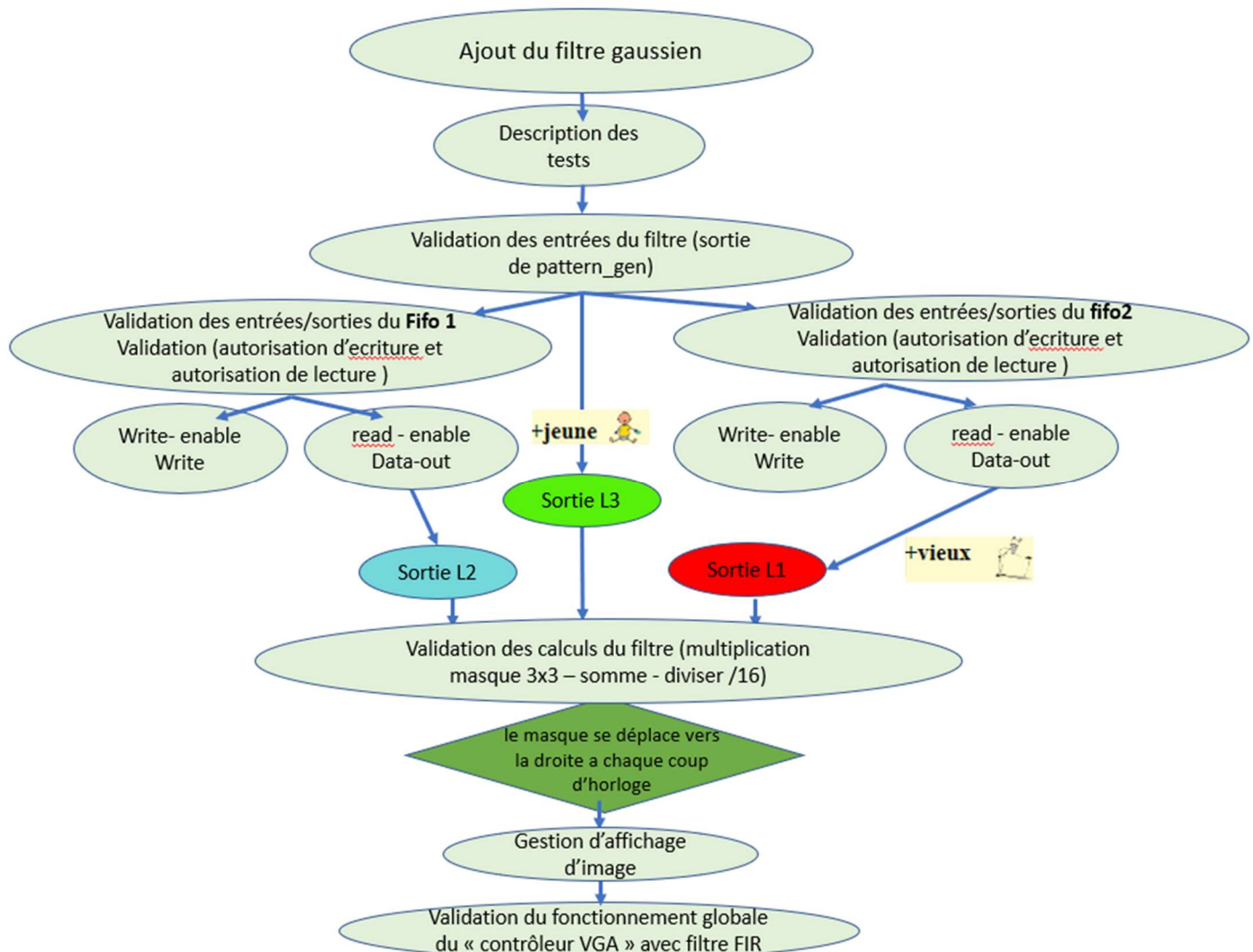


Figure 3 : plan de déroulement du plan de validation – Phase 2

III. Environnement de tests

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

	<p>Voir annexe du rapport de projet</p>
	<p>Le module PmodVGA de Digilent fait l'interface entre la carte FPGA et le port VGA de l'écran. Il comprend 14 broches d'entrée, dont 4 broches pour chaque composante Rouge-Vert-Bleu, soit 4 bits par composante ce qui confère au module une profondeur de $16 \times 16 \times 16 = 4096$ couleurs. Il reste 2 broches HS et VS pour les signaux de synchronisation horizontale et verticale. La conversion numérique-analogique est effectuée avec un simple réseau de résistances R-2R en conjonction avec la résistance de terminaison standard de 75Ω de l'écran VGA. Ce module peut transporter les signaux jusqu'à une fréquence de balayage pixel par pixel de 150 MHz</p>
	<p>Un câble VGA mâle-mâle.</p>

	<p>Un écran avec une entrée VGA</p>
	<p>Un oscilloscope HANTEK</p>

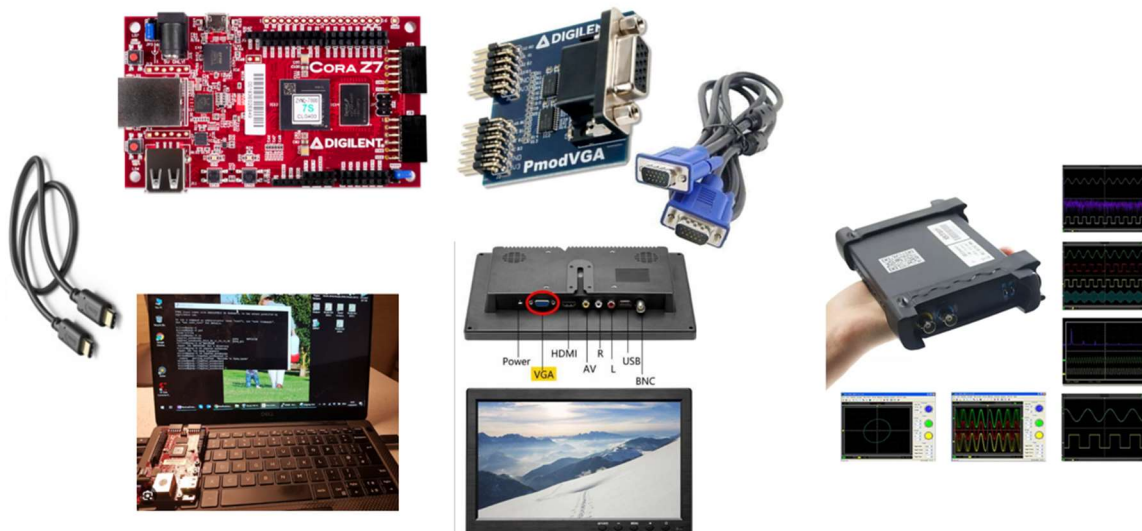


Figure 4: Outils à disposition pour réaliser les tests

IV. Description des tests phase 1 : contrôleur VGA sans filtre de GAUSS

1. Validation de la génération de l'horloge VGA « pixel clock »

Spécification :

La norme VGA 640x480 pixels à 60Hz impose une fréquence d'horloge pour les pixels de 25,2 MHz.

Explication :

- rafraîchissement de l'écran tous les $1 / 60 \text{ Hz} = 16,6 \text{ ms}$ (rafraîchi 60 fois par seconde)
- surface d'affichage $800 \times 525 \text{ pixels} = 420\,000 \text{ pixels}$
- chaque pixel doit être traité en $1 / (60 \times 800 \times 525) \approx 39,7 \text{ ns}$ (nanosecondes)
- Une horloge est donc nécessaire pour cadencer le processus de balayage pixel par pixel : à une fréquence de $1 / 39,7 \cdot 10^{-9} = 25,2 \text{ MHz} \Rightarrow$ (cela signifie qu'un pixel a une durée de $1/25200000 \text{ s}$).

Méthode / comment : observer le module ou le composant qui est responsable de cette fonction (IP- PLL par exemple) et valider les entrées / sorties (ici une horloge de 25MHz)

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 <i>clock_gen_01</i> : Vérification de la génération de « pixel_clock »		
<p><u>Résumé:</u> Tests permettant de s'assurer que la fréquence générée est celle attendue.</p> <p><u>Quoi :</u> la fréquence d'horloge d'affichage des pixels VGA est de (25,2 MHz) : cela signifie qu'un pixel a une durée de $1/25200000 \text{ s}$.</p> <p><u>comment/pré-requis :</u> Réaliser un testbench de la fonction pixel-clock avec en entrée une clock de 100MHz et une sortie pixel_clock</p>		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Lancer la simulation	<p>Vérifier qu'il y a $1/25200000 \text{ s}$ entre 2 fronts montants de la clock de sortie pixel_clock</p> <p>$\Rightarrow \text{Pixel time} = 1 / \text{Pixel clock} = 0,04 \mu\text{s}$</p>

<u>Type d'exécution:</u>	Test_bench
<u>Durée estimée d'exéc. (en min):</u>	30 min
<u>Cahier d'exigences</u>	EXIGENCE_clock_gen_01

2. Validation de la synchronisation horizontale (H-sync)

Spécification :

La synchronisation horizontale repose **sur 4 paramètres temporels** :

- Sync pulse H_PW, qui est la durée de l'impulsion de synchronisation, et qui annonce le début d'une nouvelle ligne
- Back porch H_BP, qui correspond à une durée qui précède l'affichage de la ligne
- L'horizontal video HD, qui correspond à l'affichage effectif des pixels de la ligne
- Front porch H_FP, délai après l'affichage de la ligne avant le prochain sync pulse

Timing horizontaux attendus :

Horizontal timing (line)

Polarity of horizontal sync pulse is negative.

Scanline part	Pixels	Time [μs]
Visible area	640	25.422045680238
Front porch	16	0.63555114200596
Sync pulse	96	3.8133068520357
Back porch	48	1.9066534260179
Whole line	800	31.777557100298

General timing

Screen refresh rate	60 Hz
Vertical refresh	31.46875 kHz
Pixel freq.	25.175 MHz

Méthode / comment :

- Vérifier si les durées suivantes sont exactes :
 - H_PW : Durée de l'impulsion de synchronisation = 96 pixels x 0,04 μs = 3,813 μs
 - H_BP = 48 pixels x 0,04 μs = 1,90 μs
 - HD = 640 pixels x 0,04 μs = 25,42 μs
 - H_FP = 16 pixels x 0,04 μs = 0,635 μs

- Vérifier pour chaque front montant de clock pixel et dans le cas ou RST est égal à '1' : si $h_sync = '0'$
- Vérifier pour chaque front montant de clock pixel et dans le cas ou RST est égal à '0' : si $h_sync = '1'$ quand la position pixel $H_POS \geq (HD + H_FP)$ et $(H_POS < (HD + H_FP + H_PW))$
- Vérifier si pour chaque ligne horizontale $H_PW + H_BP + HD + H_FP = 128 + 16 + 640 + 16 = 800 \times 0,04 \mu s = 32 \mu s$ (ici $31.77 \mu s$)
- Vérifier le rafraichissement à 60Hz (tous les $16,67 \text{ ms}$)
- Vérifier si on a 525 lignes par image ($16,67/0,032$)

Test différent : changer la durée de l'impulsion de synchronisation (SP) à 128 au lieu de 96 et changer la durée (BP) qui précède l'affichage à 16 pixels au lieu de 48

→ On doit retrouver : $H_PW = 128 \text{ pixels} \times 0,04 \mu s = 5,12 \mu s$ et $H_BP = 16 \text{ pixels} \times 0,04 \mu s = 0,64 \mu s$

Outils : logiciel VIVADO-XILINX

Fiche de test :

Test N°2 fonction <i>horizontal_timing_gen_03</i> : Vérification de la synchronisation horizontale		
<p><u>Résumé</u> : Tests permettant de s'assurer que la fonction <i>horizontal_timing_gen_03</i> : génère la synchronisation horizontale</p> <p>Quoi : valider un rafraichissement de l'écran (image entière) toutes les 60 Hz</p> <p>Comment/pré-requis : Réaliser un test-bench de la fonction <i>horizontal_timing_gen_03</i> avec :</p> <p>*en entrée : <i>pixel_clock</i> et <i>RESET</i></p> <p>*en sortie : signaux logiques de synchronisation horizontale (<i>hsync</i>)</p> <p>- Lancer la simulation de la fonction <i>horizontal_timing_gen_03</i> : rajouter les signaux synchronisation horizontale (<i>hsync</i>) et le compteur <i>h_cnt</i></p>		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Vérifier le rafraichissement de l'écran (image entier) toutes les 60 Hz : Mesurer le temps entre deux images consécutives quand les compteurs (<i>h_cnt</i>) et (<i>v_cnt</i>) sont à zéro	Le temps entre deux (<i>h_cnt</i>) à 0 et (<i>v_cnt</i>) à 0 et le prochain <i>h_cnt</i> = 0 = <i>v_cnt</i> doit être = $1/60 \text{ s}$ (60Hz) Equivaut à $800 \times 525 \times 0.04 \mu s = 16.8 \text{ ms}$
2.1	Vérifier la durée <i>H_PW</i> : Durée de l'impulsion de synchronisation	Le temps doit être = $96 \text{ pixels} \times 0,04 \mu s = 3,84 \mu s$

2.2	Vérifier la plage de signal H_PW :	Vérifier pour chaque front montant de clock pixel et dans le cas ou RST est égal à '0' : si $h_sync = '1'$ si la position pixel $(HD + H_FP + H_PW) > H_POS \Rightarrow (HD + H_FP) 656 \Rightarrow H_POS \Rightarrow 751$
3	Vérifier la durée de la ligne horizontale	Vérifier si j'ai pour chaque ligne horizontale $H_PW + H_BP + HD + H_FP = 128 + 16 + 640 + 16 = 800 \times 0,04 \mu s = 32 \mu s$
4	Vérifier le reset de la synchronisation horizontale	Vérifier pour chaque front montant de clock pixel et dans le cas ou RST est à '1' : si $h_sync = '0'$
5	- Test différent : changer la durée de l'impulsion de synchronisation (SP) à 128 au lieu de 96 et changer la durée (BP) qui précède l'affichage à 16 pixel au lieu de 48	On doit retrouver : $H_PW = 128 \text{ pixels} \times 0,04 \mu s = 5,12 \mu s$ et $H_BP = 16 \text{ pixels} \times 0,04 \mu s = 0,64 \mu s$
<u>Type d'exécution:</u>	Test_bench	
<u>Durée estimée d'exéc. (en min):</u>	60 min	
<u>Cahier d'exigences</u>	EXIGENCE_horizontal_timing_gen_03	

3. Validation de la synchronisation verticale (v-sync)

Spécification :

La synchronisation verticale, est composée de **4 paramètres temporels** :

- Sync pulse V_PW, qui est la durée de l'impulsion de synchronisation, et qui annonce le début d'une nouvelle image
- Back porch V_BP, qui correspond à une durée qui précède l'affichage
- Vertical video VD, qui correspond à l'affichage effectif
- Front porch V_FP, délai après l'affichage de l'image avant le prochain sync pulse

Timing verticaux attendus

Vertical timing (frame)

Polarity of vertical sync pulse is negative.

Frame part	Lines	Time [ms]
Visible area	480	15.253227408143
Front porch	10	0.31777557100298
Sync pulse	2	0.063555114200596
Back porch	33	1.0486593843098
Whole frame	525	16.683217477656

General timing

Screen refresh rate	60 Hz
Vertical refresh	31.46875 kHz
Pixel freq.	25.175 MHz

Méthode / comment :

- Vérifier
 - Le rafraichissement de l'écran (image entière) à 60 Hz toutes les 16,67 ms
 - Le rafraichissement vertical à 31.46875 kHz

Vérifier si les durées suivantes sont exactes :

- Durée de la colonne verticale 16.68 ms
- Durée de la ligne horizontale 31.77 μ s (32 μ s)
 - $V_PW = 2 \text{ lignes} \times 32 \mu\text{s} = 64 \mu\text{s}$ (Durée de l'impulsion de synchronisation verticale)
 - $V_BP = 29 \text{ lignes} \times 32 \mu\text{s} = 928 \mu\text{s}$
 - $VD = 480 \text{ lignes} \times 32 \mu\text{s} = 15,36 \text{ ms}$
 - $V_FP = 10 \text{ lignes} \times 32 \mu\text{s} = 320 \mu\text{s}$
- Vérifier l'affichage vertical = $V_PW + V_BP + VD + V_FP = 521 \times 32\mu\text{s} = 16,67\text{ms} = 1/60\text{Hz}$
- Vérifier le rafraichissement à 60Hz toutes les 16,67 ms

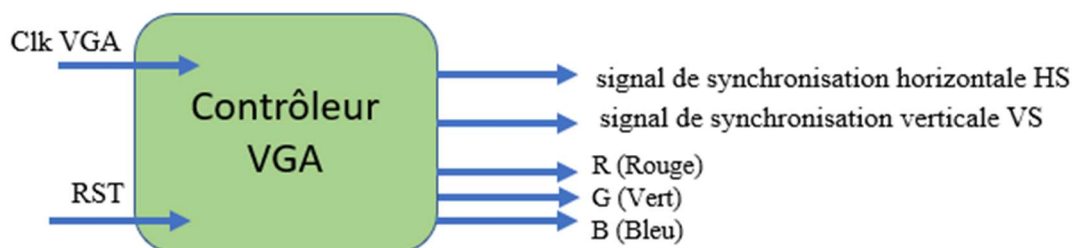
Outils : logiciel VIVADO-XILINX**Fiche de test :**

Test N°3 fonction : vertical_timing_gen_04 : Vérification de la synchronisation verticale		
<p><u>Résumé:</u> Tests permettant de s'assurer que la fonction vertical_timing_gen_04 : génère la synchronisation verticale</p> <p>Quoi : valider un rafraichissement vertical : rafraichissement d'une ligne entière à 31.46875 kHz</p> <p>comment/pré-requis : Réaliser un test-bench de la fonction vertical_timing_gen_04 avec :</p> <p>*en entrée : pixel_clock et RESET</p> <p>*en sortie : signaux logiques de synchronisation verticale (vsync)</p> <p>- Lancer la simulation de la fonction vertical_timing_gen_04 : rajouter les signaux synchronisation verticale (vsync) et le compteur v_cnt</p>		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Vérifier le rafraichissement vertical : donc rafraichissement d'une ligne complète à 31.46875 kHz quand les compteurs sont à zéro	Le temps entre deux (h_cnt) à 800 consécutifs doit être 32μs (h_cnt = fin de la ligne et v_cnt = 2 par exemple)
2.1	Vérifier la durée V_PW : Durée de l'impulsion de synchronisation verticale	Le temps doit être = 2 lignes x 32 μs = 64 μs (Durée de l'impulsion de synchronisation verticale)
2.2	Vérifier la plage de signal V_PW :	Vérifier pour chaque front montant de clock pixel et dans le cas ou RST est égal à '0' : si V_sync = '1' quand la position pixel (HD + V_FP + V_PW) > V_POS >= (VD + V_FP) Soit 490 >= V_POS >= 491
3	Vérifier la durée d'une trame verticale	Vérifier si affichage vertical = V_PW + V_BP + VD + V_FP = 521 lignes x 32μs = 16,67ms = 1/60Hz
4	Vérifier le reset de la synchronisation verticale	Vérifier pour chaque front montant de clock pixel et dans le cas ou RST est à '1' : si v_sync = '0'
<u>Type d'exécution:</u>	Test_bench	

<u>Durée estimée</u> <u>d'exéc. (en min):</u>	60 min
<u>Cahier d'exigences</u>	EXIGENCE_vertical_timing_gen_04

4. Validation de l'ensemble HSYNC et VSYNC : timing générale (timing_gen)

Spécification :

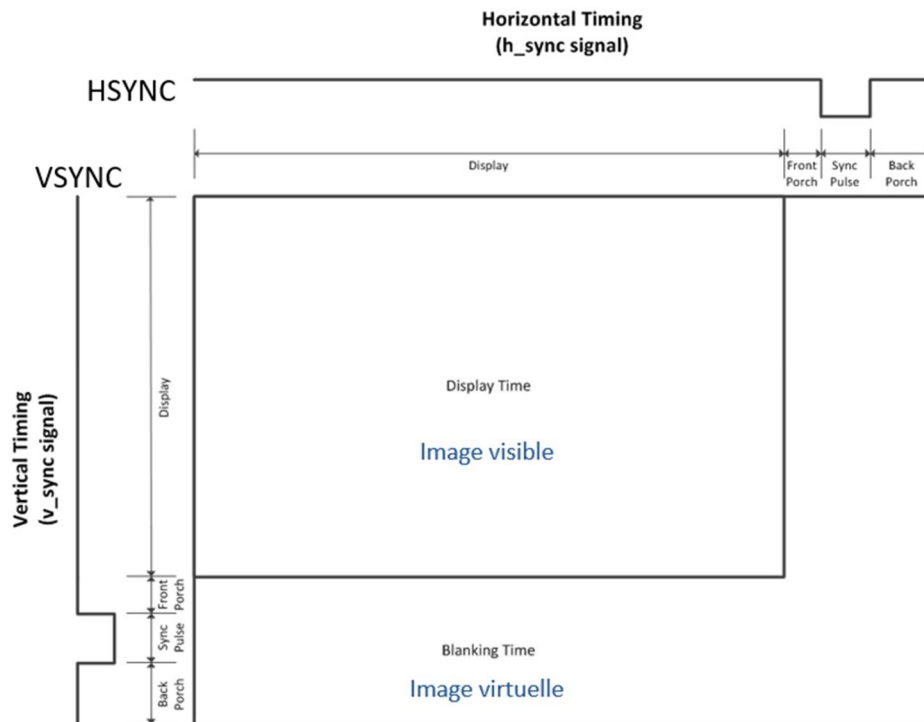


Les entrées de ce module sont :

- Un signal d'horloge pxl_clk à 25,2 MHz, fréquence de balayage de l'écran pixel par pixel ;
- Un signal RST de réinitialisation, et qui pourra être activé sur appui d'un bouton-poussoir intégré en surface de la carte (pour nous ici bouton 0).

En sortie :

- Les signaux logiques de synchronisation horizontale et verticale hsync et vsync ;
- Une sortie couleur



Méthode / comment :

Valider les entrées : (fait précédemment)

- La bonne fréquence en entrée (25.2 MHZ)
- RST actif (quand bouton appuyé)

Valider les sorties :

- Les signaux logiques de synchronisation horizontale et verticale (hsync et vsync)
- Une sortie couleur

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°4 fonction : *timing_gen_02* : Vérification de la génération des signaux

Résumé : Tests permettant de s'assurer que le composant controleur_VGA génère bien les signaux logiques de synchronisation horizontale et verticale (h_sync et v_sync)

comment/pré-requis : Réaliser un test Bench de la fonction timing_gen_02 avec :

<i>*en entrée : pixel_clock et RST</i>		
<i>*en sortie : signaux logiques de synchronisation horizontale et verticale (h_sync et v_sync) et signal couleurs</i>		
<i>- Lancer la simulation de la fonction timing_gen_02 : rajouter les signaux synchronisation horizontale (h_sync) et verticale (v_sync) ainsi que les compteurs h_cnt et v_cnt</i>		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Sur front montant de la synchronisation horizontale (h_sync), observer h_cnt.	Vérifier (position pixel) pour un front montant de h_cnt donné => doit être 656
2	Sur front descendant de la synchronisation horizontale (h_sync), observer h_cnt.	Vérifier (position pixel) pour un front descendant de h_cnt donné => doit être 751
3	Sur front montant de la synchronisation verticale (v_sync) observer h_cnt.	Vérifier (position pixel) pour un front montant de v_cnt donné => doit être 490
4	Sur front descendant de la synchronisation verticale (v_sync) observer h_cnt.	Vérifier (position pixel) pour un front descendant de v_cnt donné => doit être 491
<u>Type d'exécution :</u>	Test_bench	
<u>Durée estimée d'exéc (en min) :</u>	60 min	
<u>Cahier d'exigences</u>	EXIGENCE_timing_gen_02	

5. Validation de la position du pixel (h_cnt) et son reset

Spécification :

Les coordonnées du point en cours de la surface balayée sur la ligne (entre 0 et 799) sur 10 bits évoluant selon le sens de balayage à la fréquence de l'horloge 25,2 MHz.

Méthode / comment :

- Vérifier le reset de compteur de position Horizontale du pixel :
 - o Quand (H_cnt = 800) alors la valeur courante du compteur s'arrête de s'incrémenter et est remis à 0 ; en même temps, on réalisera l'affectation à 1 du signal d'activation du compteur vertical (V_cnt = 1).

- sinon, le compteur est incrémenté et le signal d'activation du compteur vertical est mis à 0
- Vérifier les coordonnées du point h_POS de la surface balayée (entre 0 et 799)

Observer h_POS entre ces intervalles

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 : compteur_H_POS_05 : Vérification de la position pixel sur la ligne horizontale		
<u>Résumé</u> : Tests permettant de s’assurer que la fonction compteur_H_POS compte correctement la position pixel sur la ligne horizontale		
<u>Comment/pré-requis</u> : Réaliser un test-bench de la fonction compteur_H_POS avec :		
*en entrée : pixel_clock et RESET		
*en sortie : signaux logiques de synchronisation (hsync)		
<u>N° d’étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Vérifier la position d’un pixel sur la ligne horizontale	Regarder un pixel sur ligne avec une couleur donnée : h_cnt = 355 et couleur rouge On doit voir une colonne horizontale rouge d’épaisseur 1 pixel
2	Vérifier le reset du compteur horizontal	Vérifier pour chaque front montant de clock pixel et dans le cas où RST est à ‘1’ : si le compteur H se réinitialise = ‘0’
<u>Type d’exécution</u> :	Test_bench	
<u>Durée estimée d'exéc (en min)</u> :	60 min	
<u>Cahier d’exigences</u>	EXIGENCE_compteur_H_POS_05	

6. Validation de la position du pixel sur une trame (v_cnt) et son reset

Spécification :

Les coordonnées du point en cours de la surface balayée sur la ligne verticale (entre 0 et 524) sur 10 bits évoluant selon le sens de balayage à la fréquence de l'horloge 25,2 MHz.

Méthode / comment :

- Vérifier le reset de compteur de position verticale du pixel :
 - Quand (V_cnt >= 525) et (H_cnt >= 800) alors la valeur courante du compteur s'arrête de s'incrémenter et est remis à 0 ;
 - Quand (H_cnt >= 800) le compteur vertical s'incrémente V_cnt + 1
- Vérifier les coordonnées du point V_POS de la surface balayée (entre 0 et 524)

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.

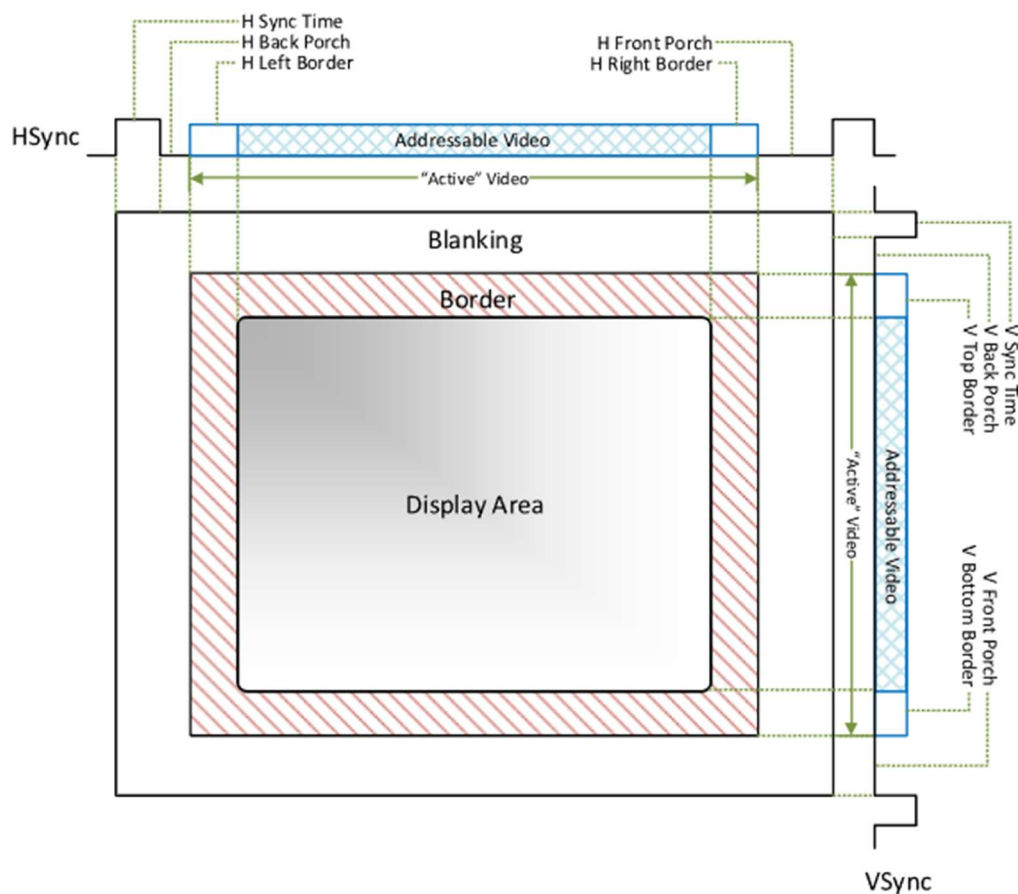
Fiche de test :

Test N°6 fonction : compteur_V_POS_06 : Vérification de la position pixel sur la ligne Verticale		
<i>Résumé : Tests permettant de s'assurer que la fonction compteur_V_POS_06 compte correctement la position pixel sur la ligne horizontale</i>		
<i>Comment/pré-requis : Réaliser un test-bench de la fonction compteur_V_POS_06 avec :</i>		
<i>*en entrée : pixel_clock et RESET</i>		
<i>*en sortie : signaux logiques de synchronisation (v_sync)</i>		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Vérifier la position d'un pixel sur la ligne Verticale	<i>Regarder un pixel sur ligne avec une couleur donnée : exemple v_cnt = 500 et couleur rouge On doit voir une ligne horizontale rouge d'épaisseur 1 pixel</i>
2	Vérifier le reset du compteur Verticale	<i>Vérifier pour chaque front montant de clock pixel et dans le cas ou RST est à '1' : si le compteur V se réinitialise à '0'</i>

<u>Type d'exécution :</u>	<i>Test_bench</i>
<u>Durée estimée d'exéc (en min) :</u>	<i>60 min</i>
<u>Cahier d'exigences</u>	<i>EXIGENCE_compteur_V_POS_06</i>

7. Validation de la zone d'affichage de l'écran (Video_On)

Spécification :



Méthode / comment :

Vérifier et valider l'état de l'écran dans ces cas :

- La sortie VIDEO_ON est à 1 quand :

- La valeur du compteur horizontal est comprise entre H_BP et H_FP
- La valeur du compteur vertical est comprise entre V_BP et V_FP
- Sinon : la sortie VIDEO_ON (active dans le code) est à 0

Vérifier si on a 525 lignes par image (16,67/0,032, dont 480 visibles)

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	Regarder le nombre de pixels visibles sur une ligne de l'écran = on doit trouver h_cnt = 640
2	Vérifier la hauteur de la zone visible	Regarder le nombre de pixels visible en hauteur sur l'écran = on doit trouver v_cnt = 480
<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	

8. Validation globale de la phase 1 « controleur_VGA » sans filtre

Test N°8 Validation globale de la phase 1 « controleur_VGA » sans filtre		
<u>Résumé</u> : Tests permettant de s’assurer de fonctionnement global de la phase 1		
<u>Comment/pré-requis</u> : observation visuelle avec :		
*en entrée : pixel_clock et RST		
*en sortie : signaux logiques de synchronisation (h_sync et v_sync avec leurs compteurs respectives)		
N° d’étape	Actions de pas:	Résultats attendus:
1	Vérifier à l’écran visuellement le bon nombre des carreaux du damier	Regarder à l’écran visuellement le bon nombre des carreaux du damier
2	Vérifier si le signal est stable	Regarder si aucune coupure (pas de lignes ni points noirs sur l’écran)
3	Verifier si aucun décalage sur l’ecran	Tester le cas d’un cadre qui entoure l’écran En noir -largeur de 3 pixel)
<u>Type d’exécution</u> :	Observation sur écran	
<u>Durée estimée d'exéc (en min)</u> :	15 min	
<u>Cahier d'exigences</u>	EXIGENCE_ display_area_07	

V. Description des tests- phase 2 : contrôleur VGA avec filtre de GAUSS

1. Validation des entrées / sorties du fifo1 (lecture- ecriture)

Spécification :

Entrée de la FIFO1 :

- Signal (pixel_L3) sur 12 bits (sortie de pattern_gen)
- 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 12 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)		
<p><u>Résumé :</u> Tests permettant de s'assurer que la fonction FIR_Filter_08 remplit son rôle de filtre de matrice 3x3</p> <p><u>Comment/pré-requis :</u> Réaliser un test-bench de la fonction FIR_Filter_08 avec :</p> <p>*en entrée : signal sortie de pattern_gen (12 bits) et autorisation d'écriture dans le FIFO1</p> <p>*en sortie : signal data RGB : couleur sur (12 bits)</p>		
<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	<p>Vérifier les conditions de lecture de la fifo :</p> <p>On lit dans FIFO_1 si :</p> <ul style="list-style-type: none"> - 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 = 800)
<u>Type d'exécution :</u>	Test_bench	

<u>Durée estimée</u> <u>d'exéc (en min) :</u>	60 min
<u>Cahier d'exigences</u>	EXIGENCE_ FIR_Filter_08

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

Spécification :

Entrée de la FIFO2 :

- Signal (pixel-L3) sur 12 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 12 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 :

Test N°10 test des entrées / sorties du fifo2 (lecture- écriture)		
<u>Résumé :</u> Tests permettant de s'assurer que la fonction FIR_Filter_08 remplit son rôle de filtre de matrice 3x3		
<u>Comment/pré-requis :</u> Réaliser un test-bench de la fonction FIR_Filter_08 avec :		
*en entrée : signal sortie de pattern_gen (12 bits) et autorisation d'écriture dans le FIFO1		
*en sortie : signal data RGB : couleur sur (12 bits)		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>

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 = 800
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 = 800
<u>Type d'exécution :</u>		Test_bench
<u>Durée estimée d'exéc (en min) :</u>		60 min
<u>Cahier d'exigences</u>	EXIGENCE_ FIR_Filter_08	

3. Validation des entrées / sorties du Conv-filter-Gauss (calcul)

Spécification :

Entrée de Conv-filter-Gauss:

- Signal (pixel-L3) sur 12 bits (sortie de pattern_gen)
- Signal sortie de d-out (pixel-L2) du FIFO 1 sur 12 bits
- Signal sortie de d-out (pixel-L1) du FIFO 2 sur 12 bits

Sortie de Conv-filter-Gauss:

- Signal pixel-out : sur 12 bits

Méthode / comment :

Nous avons un masque kernel 3x3 : K =	Nous avons un masque de Gaussien 3x3:	Les pixels utilisés dans le calcul sont nommés
C1 C2 C3	1 2 1	C1 C2 C3
L1 K11 K12 K13	2 4 2	L1 P11 P12 P13
L2 K21 K22 K23	1 2 1	L2 P21 P22 P23
L3 K31 K32 K33		L3 P31 P32 P33

Etant donné que tous les coef sont des multiples de 2, nous allons utiliser les décalages de bits pour les opérations de multiplication et de division.

A chaque coup d'horloge, on décale de 1 le masque vers la droite, donc les pixels à considérer vont vers la gauche : $P_{11} \leq P_{12}$, $P_{21} \leq P_{22}$, ... $P_{32} \leq P_{33}$

Les valeurs de P13, P23 et P33 sont les nouvelles valeurs à considérer, donc les inputs de la fonction Pixel_L1 / Pixel_L2 / Pixel_L3

Pixel_out sera la valeur du pixel à appliquer aux coordonnées du pixel P22 en sortie du filtre (pour pixel_out nous avons besoin d'un signal 16 bits car $\max = (\text{signal } 12 \text{ bits}) * (4 \text{ bits})$ (poids masque 16 = 2^4)

Outils : logiciel VIVADO-XILINX

Fiche de test :

Test N°11 test des entrées / sorties du Conv-filter-Gauss (calcul)		
<p><u>Résumé :</u> Tests permettant de s'assurer que la fonction FIR_Filter_08 remplit son rôle de filtre de matrice 3x3</p> <p><u>Comment/pré-requis :</u> Réaliser un test-bench de la fonction FIR_Filter_08 avec :</p> <p>*en entrée : signal sortie de pattern_gen (12 bits) et autorisation d'écriture dans le FIFO1</p> <p>*en sortie : signal data RGB : couleur sur (12 bits)</p>		
<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Vérifier la première opération de calcul : multiplication : Application du masque KERNEL	Regarder le décalage de bit à droite en fonction des éléments du masque (x2 = décalage 1 bit à droite) (x4 = décalage 2 bits à droite)
2	Vérifier la somme	Vérifier la somme

3	Vérifier la 3ème opération de calcul : division	Regarder le décalage de bit à gauche : on divise par le poids du masque (16) (/16 = /2^4 donc on aura un décalage 4 bits à gauche)
<u>Type d'exécution :</u>	Test_bench	
<u>Durée estimée d'exéc (en min) :</u>	30 min	
<u>Cahier d'exigences</u>	EXIGENCE_ FIR_Filter_08	

4. Validation de la synchronisation du système après l'ajout du filtre

Spécification :

L'ajout de deux FIFOs implique l'ajout d'une latence à l'ensemble du projet.

Méthode / comment :

- Prendre en compte le facteur latence des FIFOs
- Se synchroniser avec HSYNC et VSYNC pour un affichage sur écran

Outils : logiciel VIVADO-XILINX

Fiche de test :

Test N°12 de la synchronisation du système après l'ajout du filtre		
<u>Résumé :</u> Tests permettant de s'assurer que la fonction FIR_Filter_08 rempli son role de filtre de matrice 3x3		
<u>Comment/pré-requis :</u> Réaliser un test-bench de la fonction FIR_Filter_08 avec :		
*en entrée : signal sortie de pattern_gen (12 bits) et autorisation d'écriture dans le FIFO1		
*en sortie : signal data RGB : couleur sur (12 bits)		
<u>N° d'étape</u>	<u>Actions de pas :</u>	<u>Résultats attendus :</u>
1	Vérifier les coordonnées h_sync et v_sync	Regarder les coordonnées h_sync et v_sync
2	Regarder le nombre de cycle d'horloge de décalage	Regarder les signaux h_decal et v_decal sont ajoutés aux calculs de h_sync et v_sync

<u>Type d'exécution :</u>	<i>Test_bench</i>
<u>Durée estimée d'exéc (en min) :</u>	<i>40 min</i>
<u>Cahier d'exigences</u>	<i>EXIGENCE_display_area_07, timing_gen_02</i>

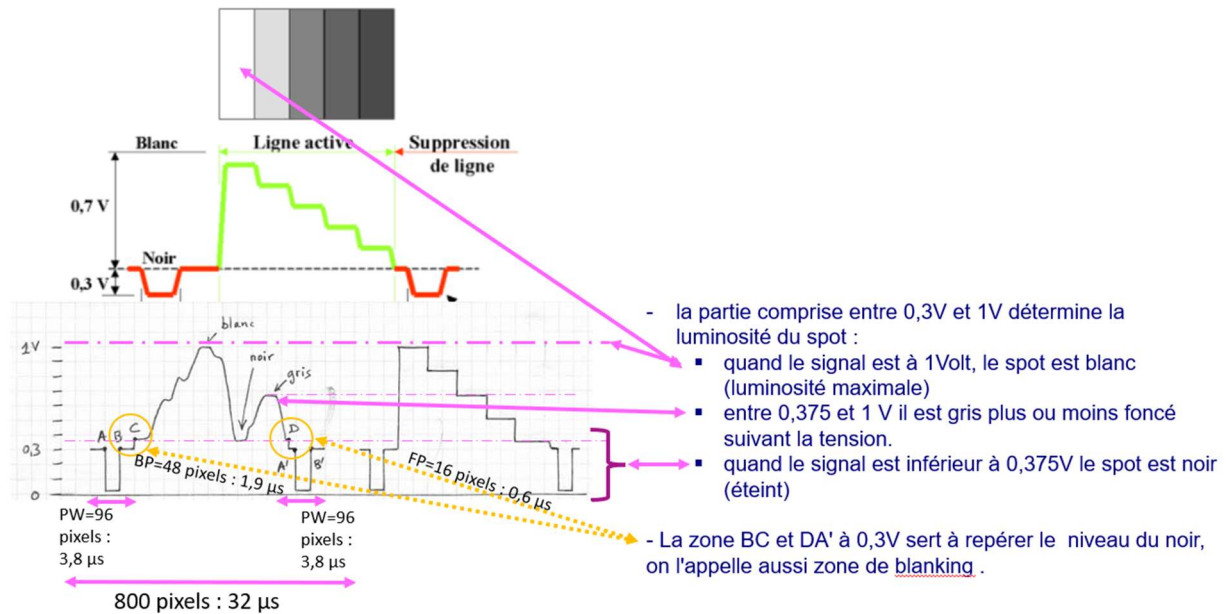
5. Validation du dégradé de gris dans les bords après l'ajout du filtre

Spécification :

La palette de notre image en niveaux de gris est constituée par les 16 couleurs comprises entre 0,0,0 et 15,15,15 donc **16 nuances de gris**

Méthode / comment :

- Nous avons deux couleurs noir et blanc qui varient de 0 à 1 Volt
- La partie comprise entre 0,3V et 1V détermine la luminosité du spot :
 - Quand le signal est à 1Volt, le spot est blanc (luminosité maximale)
 - Entre 0,375 et 1 V il est gris plus ou moins foncé suivant la tension.
 - Quand le signal est inférieur à 0,375V le spot est noir (éteint)
- La zone BC et DA' à 0,3V sert à repérer le niveau du noir, on l'appelle aussi zone de blanking.
- La ligne représentée à droite, dite en escalier, commence par une plage blanche puis par un gris clair puis par une plus foncée, pour arriver jusqu'au noir ; si toutes les lignes de l'image sont identiques à celle-ci, on obtient une mire de barres.



Outils : logiciel VIVADO-XILINX et carte CORA Z7

Fiche de test :

Test N°13 de la dégradation de gris dans les bords après l'ajout du filtre		
Résumé: Tests permettant de s'assurer que la palette de notre image en niveaux de gris est constituée par les 16 couleurs comprises entre 0,0,0 et 15,15,15 qui y sont sauvegardées donc 16 nuances de gris		
N° d'étape	Actions de pas :	Résultats attendus :
1	Vérifier le nombre de dégradation de la couleur gris	Regarder le niveau de dégradé de gris entre un carreau blanc et noir (fff et 000 en hexa)
2	Regarder le nombre de cycle d'horloge de décalage	Regarder les signaux h_decal et v_decal sont ajoutés aux calculs de h_sync et v_sync
Type d'exécution: Test_bench		
Durée estimée d'exéc (en min) : 40 min		
Cahier d'exigences EXIGENCE_ horizontal_ timing_gen_03 et vertical_ timing_gen_04		

6. Validation globale de la phase 2 « controleur_VGA » avec filtre

Test N°14 Validation globale de la phase 2 « controleur_VGA » avec filtre		
<u>Résumé</u> : Tests permettant de s’assurer de fonctionnement global de la phase 2		
<u>Comment/pré-requis</u> : observation visuelle 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 à l’écran visuellement le bon nombre des carreaux du damier	Regarder à l’écran visuellement le bon nombre des carreaux du damier
2	Vérifier si le signal est stable	Regarder si aucune coupure (pas de lignes ni points noirs sur l’écran)
3	Verifier si aucun décalage sur l’ecran	Tester le cas d’un cadre qui entoure l’écran En noir -largeur de 3 pixel)
<u>Type d’exécution</u> :	Observation sur écran	
<u>Durée estimée d'exéc (en min)</u> :	15 min	
<u>Cahier d’exigences</u>	EXIGENCE_ display_area_07	

VI. Moyen de test 2 :test sur cible-set up-debug- ILA

Test fonctionnel sur cible

Test N°15 sur cible horizontal_timing_gen_03 : Vérification de la synchronisation horizontale		
<u>Résumé</u> : Tests permettant de s'assurer que la fonction horizontal_timing_gen_03 génère bien les signaux logiques de synchronisation horizontale		
<u>Quoi</u> : validation sur cible de signal de synchronisation horizontale (hsync)		
<u>Comment/pré-requis</u> : Réaliser un test ILA de la fonction horizontal_timing_gen_03		
<u>N° d'étape</u>	<u>Actions de pas :</u>	<u>Résultats attendus :</u>

1	Lancer la simulation de la synchronisation horizontale (hsync)	Mettre un trigger sur un front montant de la sortie de VGA-HS-OBUFF → Observer le décalage de registre h_cnt-reg pour assurer la synchronisation
2	Vérifier le rafraichissement de l'écran (image entier) à 60 Hz	Regarder le temps entre deux triggers placés sur h_cnt=0 et v_cnt=0 (Le temps doit être = 1/60 s)
<u>Type d'exécution :</u>	ILA (sur cible)	
<u>Durée estimée d'exéc (en min) :</u>	30 min	
<u>Cahier d'exigences</u>	EXIGENCE_ horizontal_ timing_gen_03	

Test fonctionnel sur cible

Test N°16 test fonctionnel sur cible vertical_ timing_gen_04 : Vérification de la synchronisation verticale		
<u>Résumé :</u> Tests permettant de s'assurer que la fonction vertical_timing_gen_04 génère bien les signaux logiques de synchronisation verticale <u>Quoi :</u> validation sur cible de signal de synchronisation horizontale (vsync) <u>Comment/pré-requis :</u> Réaliser un test ILA de la fonction vertical_timing_gen_04		
<u>N° d'étape</u>	<u>Actions de pas :</u>	<u>Résultats attendus :</u>
1	Lancer la simulation de la synchronisation verticale (vsync)	Mettre un trigger sur un front montant de la sortie de VGA-VS-OBUFF → Observer le décalage de registre h_cnt-reg pour assurer la synchronisation
2	Vérifier le rafraichissement vertical à 31.46875 kHz	Regarder le temps entre deux triggers placés sur h_cnt=800 et v_cnt=2 (Le temps doit être = 1/31.46875 kHz)
<u>Type d'exécution :</u>	ILA (sur cible)	

<u>Durée estimée</u> <u>d'exéc (en min) :</u>	30 min
<u>Cahier d'exigences</u>	EXIGENCE_ vertical_ timing_gen_04

Test fonctionnel sur cible**Test N°17** Vérification de la génération des signaux, sortie de calcul et sorties des FIFOs

Résumé : Tests permettant de s'assurer que le composant controleur_VGA génère bien les signaux logiques de synchronisation horizontale et verticale (hsync et vsync) et le bon calcul après ajout de filtre.

Quoi : validation sur cible des signaux logiques de synchronisation horizontale et verticale (hsync et vsync)

Comment/pré-requis : Réaliser un test ILA de la fonction timing_gen_02, FIR_Filter_08,

<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Vérifier la synchronisation horizontale (hsync), avec la sortie de calcul du filtre et les sorties des Fifo	Mettre un trigger sur un front montant : FIFO1 read-enable et FIFO2 write-enable en regardant le h-sync
2	Vérifier la sortie du filtre (calcul)	Mettre un trigger sur les sorties des PIXEL L1 PIXEL L2 et PIXEL L3
<u>Type d'exécution :</u>	ILA (sur cible)	
<u>Durée estimée</u> <u>d'exéc (en min) :</u>	60 min	
<u>Cahier d'exigences</u>	EXIGENCE_ timing_gen_02 , FIR_Filter_08	

Test N°18 ILA test des entrées / sorties du Conv-filter-Gauss (calcul)

Résumé : Tests permettant de s'assurer que la fonction FIR_Filter_08 remplit son rôle de filtre de matrice 3x3

Comment/pré-requis : Paramétrer l'ILA avec les signaux à observer et à trigger.

<u>N° d'étape</u>	<u>Actions de pas:</u>	<u>Résultats attendus:</u>
1	Vérifier les conditions d'écriture dans la FIFO_1 et 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 le flux de l'autorisation de lecture de la FIFO_1 et FIFO_2	Regarder (FIFO_2_read_ena) est active : quand la deuxième ligne est entièrement lue
<u>Type d'exécution :</u>	Ila	
<u>Durée estimée d'exéc (en min) :</u>	30 min	
<u>Cahier d'exigences</u>	EXIGENCE_ FIR_Filter_08	

Test N°19 ILA test dégradée du gris sur ILA

Résumé : Tests permettant de voir les niveaux de gris suite à l'application du filtre de matrice 3x3

Comment/pré-requis : Paramétrer l'ILA avec les signaux à observer et à trigger.

<u>N° d'étape</u>	<u>Actions de pas :</u>	<u>Résultats attendus :</u>
1	Vérifier les niveaux de dégradés du gris	Regarder combien de Niveau de gris
<u>Cahier d'exigences</u>	EXIGENCE_ FIR_Filter_08	

VII. MOYEN DE TEST 3 : test sur cible- oscilloscope

Test N°20 Mesure du Temps entre 2 lignes

Test N°21 Mesure Temps de H_sync = 3.8μs

Test N°22 Mesure pulse H_PW

Test N°23 Mesure de V-SYNC

Test N°24 Mesure de H-SYNC

Test N°25 Mesure de dégradé du gris avec V-SYNC

Test N°26 Mesure de dégradé du gris avec H-SYNC