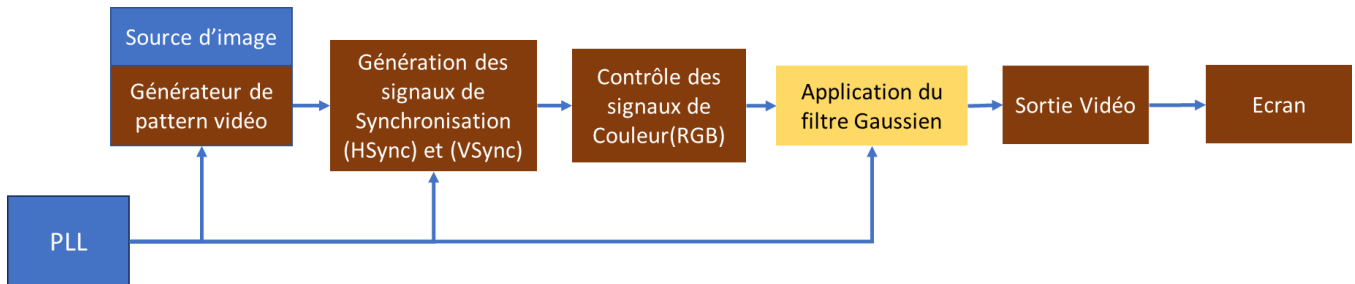
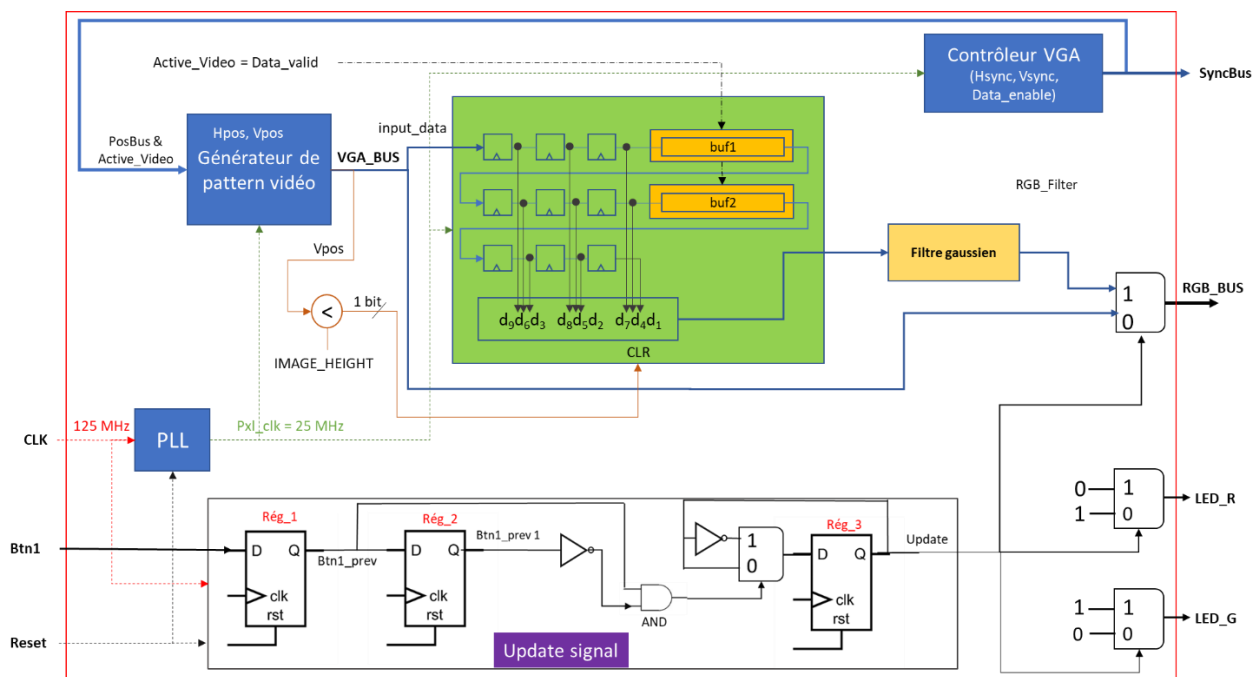


Plan de validation- Phase 2

Pour finaliser la mise en œuvre du système, nous avons ajouté le filtre gaussien à la configuration proposée initialement lors de la phase intermédiaire. La Figure ci-dessous illustre le synoptique du système final.



La Figure suivante présente l'architecture complète de notre système.



Par rapport à l'architecture précédente du système lors de la phase intermédiaire, nous avons introduit plusieurs modules supplémentaires. Parmi ces modules, nous avons utilisé des registres et des buffers pour stocker les pixels de l'image avant la convolution (d_9, d_8, \dots, d_1). Cette approche nous permet d'économiser du temps en évitant d'extraire les pixels à partir d'une mémoire externe à chaque étape de la convolution.

De plus, lorsque nous avons terminé la lecture de tous les pixels actifs, ce qui correspond à un V_{pos} supérieur à la hauteur de l'image "Image_Height", les buffers et les registres sont vidés de leurs données. Le stockage et le décalage des données dans les buffers commencent uniquement lorsque le signal "Data_valide" est égal à 1, c'est-à-dire lorsque "Data_valide" correspond à "Active_video". Cette stratégie nous permet de traiter l'image en faisant passer les pixels à travers une fenêtre plutôt que de déplacer le noyau sur l'image. Cette architecture, largement utilisée, est caractérisée par deux buffers accompagnés de quelques registres de mémoire, facilitant ainsi le chargement d'un voisinage 3×3 lors de la convolution. Elle offre une solution efficace pour la réalisation de l'opération de convolution 2D.

Le module "Filtre Gaussien" est utilisé pour appliquer l'opération de convolution, qui nécessite 9 opérations de multiplication et 8 opérations d'addition.

Dans notre architecture, nous avons exploité nos travaux pratiques récents pour développer une architecture conviviale en introduisant un module supplémentaire « Update signal » (qui n'est pas essentiel pour le traitement). Ce module permet

aux utilisateurs d'activer ou de désactiver le filtre gaussien et allume une LED correspondante en fonction de leur choix. Dans cet exemple, lorsque le filtre gaussien est activé, une LED verte s'allume ; sinon, une LED rouge s'allume.

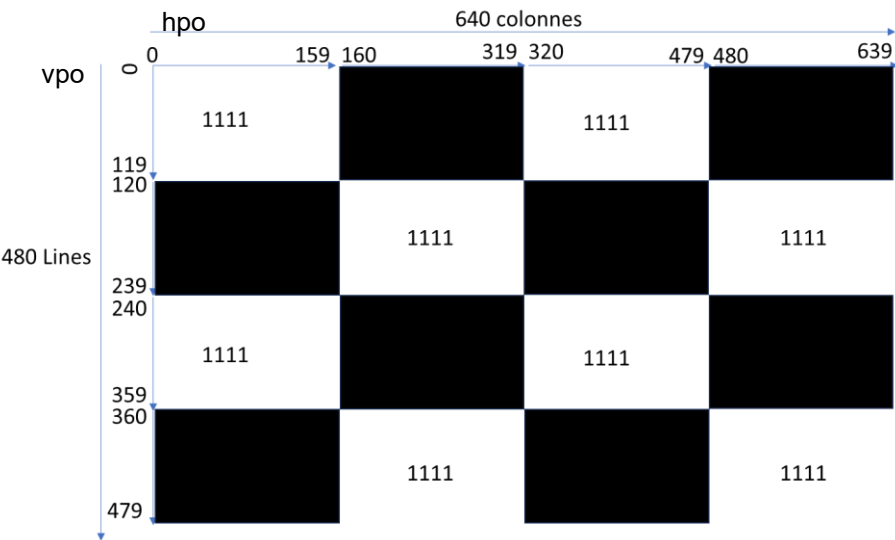
L'activation ou la désactivation du filtre gaussien se produit exclusivement lors de la réception du signal "update". Cela se produit lorsque le système détecte le premier front montant du signal provenant du bouton_1. Le signal "update" ne change de valeur que lorsque le bouton_1 est enfoncé à nouveau, c'est-à-dire lors de la détection d'un nouveau front montant. Il est important de noter que maintenir le bouton enfoncé ne maintient pas le signal "update" à sa valeur actuelle de façon continue.

1- Test de fonctionnalité :

Vérifiez si le système peut effectuer correctement la convolution 2D en utilisant des images de test prédéfinies et des noyaux de filtre connus.

Vous pouvez suivre les étapes suivantes :

- i- Créez une image de test prédéfinie comprenant 4 lignes et 4 colonnes, avec des carreaux alternant entre noir et blanc. Assurez-vous que les dimensions de l'image sont compatibles avec la norme VGA. Soit l'image/pattern ci-dessous à générer.



numéro de colonne = position horizontale = hpos

numéro de ligne = position verticale = vpos

- ii- Vérifiez si l'application de la convolution 2D avec l'architecture de la fenêtre coulissante sur une image composée de carrés noirs et blancs produit des valeurs de pixels entre les niveaux de couleur 1111 et 0000.

Pour rappel, dans le cas du filtrage de gauss, le cœur de convolution représenté ci-dessous :

1	2	1
2	4	2
1	2	1

Lorsque nous avons appliqué le produit de convolution à notre image de test prédéfinie, nous allons observer des lignes intermédiaires entre le noir et le blanc. Cette observation est expliquée par les effets de la convolution sur les transitions entre les zones noires et blanches de l'image, résultats qui ont été vérifiés par calcul théorique (cf résultat dans tableau ci-dessous).

En commentant ces résultats, nous soulignons que la convolution est un processus de filtrage qui lisse les transitions entre les pixels adjacents. Dans notre cas, les lignes et les colonnes intermédiaires que nous observons sont le résultat de ce lissage, créant une gradation subtile entre les zones noires et blanches. Cela contribue à une apparence plus douce et plus cohérente de l'image traitée.

Pour examiner cette gradation, nous pouvons observer les pixels qui se trouvent sur une transition horizontale ou verticale entre blanc et noir (ou inversement).

Pour pouvoir observer cette transition à l'oscilloscope, nous proposons d'observer les pixels qui se trouvent sur une transition horizontale, à Vpos allant par exemple de 359 à 362, et pour chaque ligne, hpos à 80, 240, 400 et 560 . Soit les valeurs attendue dans le tableau suivant, avec :

- input_data = valeur du pixel avant convolution
- output_data= pixel après convolution :

vpos	359				360				361			
hpos	80	240	400	560	80	240	400	560	80	240	400	560
input_data	1111	0000	1111	0000	0000	1111	0000	1111	0000	1111	0000	1111
output_data	1111	0000	1111	0000	1011	0011	1011	0011	0011	1011	0011	1011

vpos	362			
hpos	80	240	400	560
input_data	0000	1111	0000	1111
output_data	0000	1111	0000	1111

2- Test à l'oscilloscope :

- Préparez des images avant et après l'application de la convolution 2D en utilisant l'architecture de la fenêtre coulissante.
- Connectez une sonde de l'oscilloscope à la sortie du signal hsync du connecteur « **Pmod VGA** » et affichez la courbe correspondante sur l'écran de l'oscilloscope sur la voie 1.
- En parallèle, placez une deuxième sonde de l'oscilloscope sur la sortie d'un signal couleur du connecteur « **Pmod VGA** » (par exemple, VGA_rouge) et affichez la courbe correspondante sur l'écran de l'oscilloscope sur la voie 2. Observer alors à l'oscilloscope plusieurs périodes de hsync (minimum 4), chaque période étant une ligne du pattern, avec le signal rouge :
 - sans transition horizontal
 - avec transition horizontale

- Comparez les résultats affichés à l'écran après l'application de la convolution 2D avec ceux calculés à l'aide du logiciel Ltspice, pour assurer la précision des résultats. Cette comparaison vous permettra de valider la performance de votre système de convolution 2D et de s'assurer de la fiabilité des résultats obtenus. Les valeurs correspondantes entre les valeurs analogiques retournées par l'oscilloscope et celles répertoriées dans le tableau présenté ci-dessous, pour une transition horizontale :

vpos	359				360				361			
hpos	80	240	400	560	80	240	400	560	80	240	400	560
output_data	1111	0000	1111	0000	1011	0011	1011	0011	0011	1011	0011	1011
tension analogique(V) théorique	3,1	0	3,1	0	2,27	0,62	2,27	0,62	0,62	2,27	0,62	2,27
Tension mesurée(V) à l'oscilloscope												

vpos	362			
hpos	80	240	400	560
output_data	0000	1111	0000	1111
Tension analogique(V) théorique	0	3,1	0	3,1
Tension mesurée(V) à l'oscilloscope				

3- Démonstration visuelle :

- Préparez des images avant et après l'application de la convolution 2D à l'aide de la « sliding window architecture ». Présentez ces images côte à côte pour mettre en évidence l'effet apporté par le filtrage : un effet de flou sur les transition entre les carreaux.