Elaboration d'une interface VME connectée à la caméra EEV-CAM17

Rapport technique - 9/1/98

Christine DREZEN

Jean-Claude LACOTTE

Daniel BOGET

LAPP Annecy, BP110

74941 Annecy le Vieux Cedex

1 Caractéristiques de la caméra EEV et du capteur CCD17.

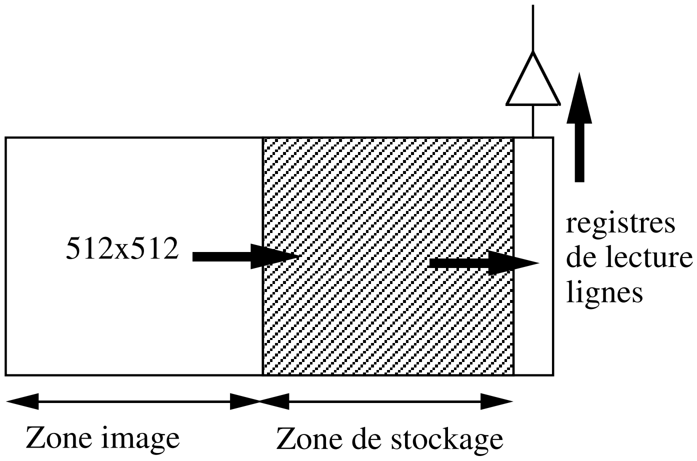


Figure 1: Capteur à transfert de trames.

Le capteur de la caméra EEV [ref.1] est un CCD à transfert de trames de 512x512 pixels ($15\mu m$ de côté par pixel), recouvrant une surface photo-sensible de 7.7x7.7 mm^2. Après une période d'intégration de la lumière, les charges photo-générées sont transférées rapidement vers une zone mémoire de stockage couverte d'un écran opaque. Un ADC convertit alors la sortie analogique des registres de lecture ligne avec une dynamique de 8 bits. L'ensemble de ces opérations constitue la période image. Il faudra ainsi prévoir sur l'interface de la mémoire pour stocker ces données numérisées.

- La caméra est aussi équipée:
  - d'un dispositif anti-blooming pour recueillir le trop plein de charges photo-générées en cas de saturation.
  - d'un réglage du temps d'intégration de la lumière.

– d'un contrôle de gain dont le rôle est d'amplifier les courants photo-générés.

Le temps d'intégration et le contrôle de gain auront la possibilité d'être définis par programmation au niveau du client graphique de GalaXie. Il faut donc organiser sur l'interface, des modules électroniques de commande de temps d'intégration et d'ajustement du gain.

- La vitesse d'acquisition maximale est de 60 images par seconde (avec une horloge de base Hb de 20 MHz). La durée d'une ligne est $32\mu s$, et celle d'une image est 16 ms. Ces paramètres de durée des signaux vidéo ne sont cependant pas conformes au standard CCIR Européen, qui impose:

  – 50 images vidéo par seconde (en non entrelacé).
  – des lignes de durée $64\mu s$ (niveau vidéo entre 0.34V et 1V).
  – une durée de 20 ms par image.
  – des signaux de synchronisation horizontale et verticale (niveau 0V), et des signaux de suppression ligne et trame (niveau 0.3V) de formats adéquats.

  Il faut donc prévoir un module de reconstruction de la vidéo qui réponde aux normes en vigueur.

2 Constitution de l'interface caméra.

2.1 Modules à prévoir sur la carte.

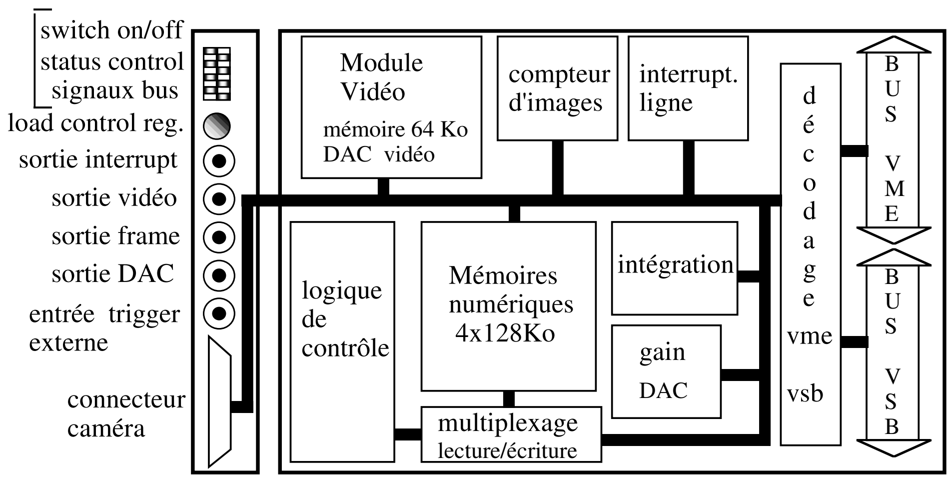


Figure 2: Carte interface caméra.

La figure 2 donne un aperçu de la carte interface avec ses différents modules, ainsi que sa face avant munie d'entrées/sorties et de boutons de contrôle et de commande. Chaque fonctionnalité est décrite plus précisément dans les paragraphes suivants.

2.2 Module de reconstruction de la vidéo.

On rappelle que la vidéo produite par la caméra présente les caractéristiques suivantes:

- 512x512 lignes de durée $32\mu s$x 512, soit 16 ms par image, soit encore 60 images par seconde.

Nous devons reconstruire une vidéo standard:

- de durée $64\mu s$ par ligne, soit 20 ms par image, soit encore 50 images par seconde.

La solution envisagée prévoit de lire un pixel sur deux, soit 256 pixels par ligne, et ceci une ligne sur deux. La reconstitution de la vidéo est une image 256x256 dont la résolution reste suffisante sur moniteur. En appliquant cette stratégie, toutes les durées des signaux sont respectées. Les données numériques brutes produites au rythme de 20 MHz sont 'triées' par une logique adéquate, et les données extraites (256x256 pixels) sont stockées dans une mémoire de 64Ko. La lecture depuis cette mémoire est cadencée par une horloge de 5 MHz, qui charge les données vers un DAC vidéo. Cet élément permet de convertir les informations numériques en données analogiques mélangées ensuite aux signaux de synchronisation et de suppression ligne et trame. La sortie de ce type de DAC donne une vidéo composite avec tous les niveaux analogiques appropriés pour chaque type de signaux. La figure 3 illustre le principe de base de reconstruction de la vidéo. Une sortie vidéo (BNC) est prévue en face avant de l'interface.

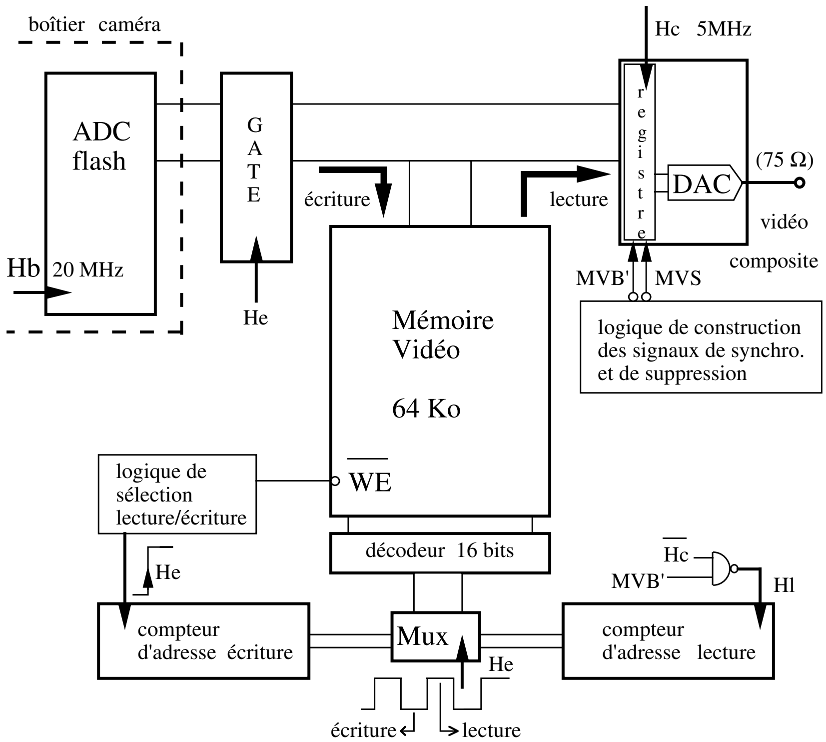


Figure 3: Principe de reconstruction de la vidéo.

Ce module pourra être implanté à l'intérieur d'un petit boîtier près de la caméra et hors interface, pour disposer localement d'une sortie vidéo.

2.2.1 Logique de sélection lecture/écriture en mémoire vidéo:

Cette logique illustrée sur la figure 4 permet d'extraire les données numériques intéressantes issues de la caméra, pour la constitution de l'image vidéo 256x256. Un premier bloc logique permet la sélection d'une ligne sur deux, l'autre bloc logique d'un pixel sur deux: Alors pour deux lignes (2 x 32μs) de 512 pixels issues de la digitisation des données dans la caméra, on extrait de la première ligne 256 pixels, alors qu'on en extrait aucun de la seconde. On a construit une ligne vidéo de 256 pixels de durée 64μs compatible avec le standard CCIR.

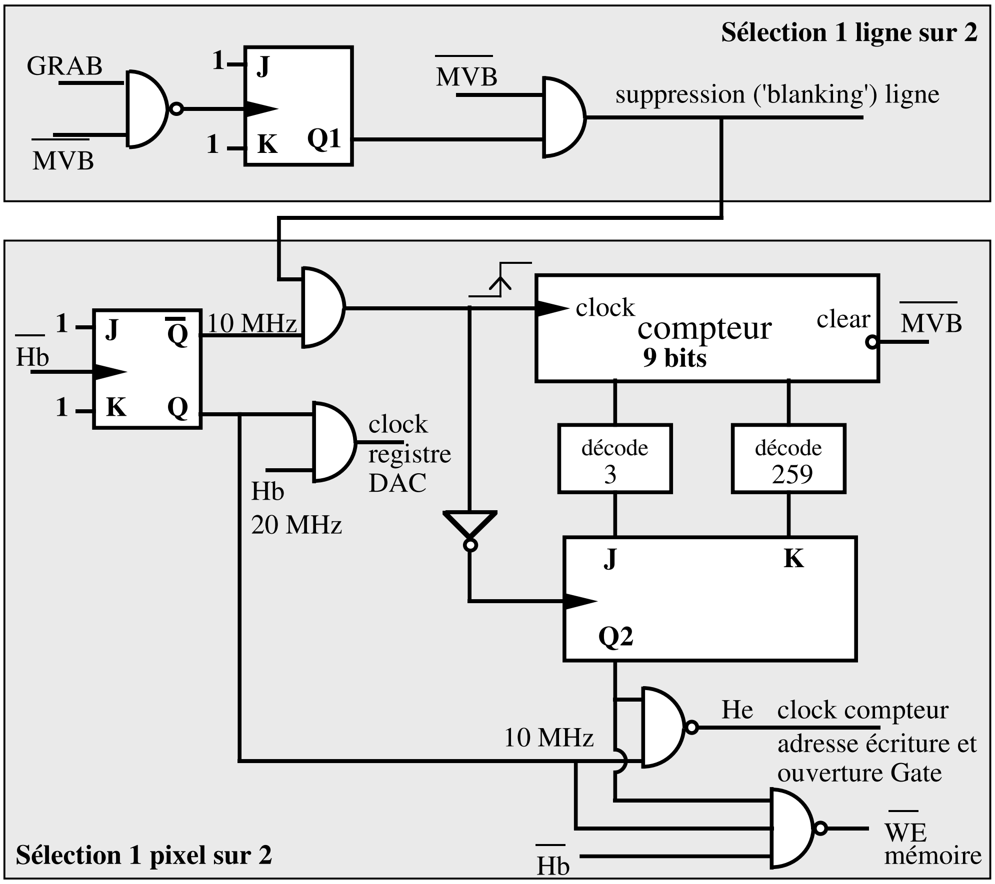


Figure 4: Logique de sélection lecture/écriture.

Principe de la logique: Pendant que sortent les pixels actifs de chaque ligne digitisée (l'horloge compteur est en effet inhibée pendant la durée des signaux de blanking car à ces instants il n'y a pas de données digitisées disponibles relatives à la zone active du CCD), on autorise une horloge de fréquence 10 MHz (la moitié de la fréquence de base de digitisation des données dans la caméra qui définit l'horloge de sortie des 512x512 pixels utiles) sur un compteur dont le décodage des sorties fournit l'encadrement (sortie Q de la bascule JK) des 256 coups d'horloge à 10 MHz pour l'ouverture de la Gate. C'est donc aussi cette même horloge qui permet d'incrémenter le compteur d'adresse écriture de la mémoire vidéo, en parallèle avec l'ouverture de la Gate qui présente une donnée pixel à stocker sur les entrées de la mémoire.

2.2.2 Logique de reconstruction des signaux de suppression et de synchronisation pour la vidéo non entrelacée:

Ces signaux sont à combiner aux données stockées en mémoire vidéo, avant leur reconversion en signal analogique pour la constitution de la vidéo composite. Leurs rôles sont de générer:

- Un signal de fin de ligne pour ramener le balayage horizontal en début de nouvelle ligne (tous les $64\mu s$).
- Un signal de fin de frame pour ramener le balayage vertical du bas de l'écran vers le haut de l'écran en début de nouvelle image (toutes les 20 ms).

Ces signaux de fin de ligne et de fin d'image sont constitués d'un niveau dit de suppression ('blanking') à 0.3V, pendant lequel se déclenche le signal de synchronisation (descente à 0V). C'est sur ce front descendant que le spot est ramené soit en début de ligne, soit en début d'image. Le signal vidéo quant à lui, varie entre 0.3V (noir) et 1V (blanc).

Il faut respecter un certain timing pour construire ces signaux suivant le standard CCIR [ref.2]. Voici les caractéristiques de nos signaux de synchronisation et de suppression sachant que le DAC lit les données stockées en mémoire vidéo à la fréquence (Hc) de l'ordre de 5 MHz (période de 202.4 ns soit 4 x 50.6 ns période de base). La figure 5 donne l'aspect de la vidéo composite issue du DAC.

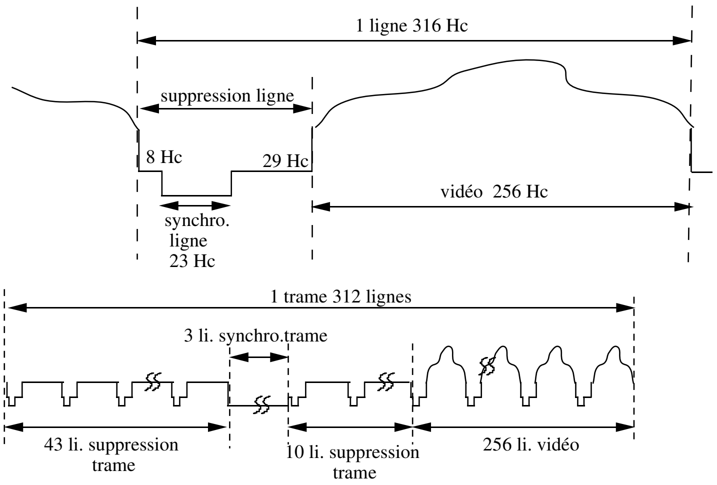


Figure 5: Timing de suppression et de synchronisation pour la vidéo.

- durée d'une ligne: $64\mu s$ soit 316xHc.
- nombre de pixel à sortir: 256xHc.
- durée de la suppression ligne: (316-256)xHc = 60xHc.
- durée de la synchro. ligne: 23xHc.
- intervalle de garde: 8xHc.
- durée d'une image: 20 ms soit la durée de 312.5 lignes.
- durée de la vidéo: 256 lignes soit 16.4 ms.

• durée de la suppression trame: (312-256) = 56 lignes réparties en 43 lignes de suppression trame + 3 lignes de synchro. trame + 10 lignes de suppression trame.

Remarques:

La trame commence en même temps que le début d'une ligne (en début de suppression ligne) et se termine en fin de la dernière ligne vidéo.

L'adresse du compteur de lecture de la mémoire vidéo doit être maintenu à zéro pendant toute la suppression trame en début de chaque image. Cette adresse ne doit pas être incrémentée pendant chaque suppression ligne. L'horloge Hl du compteur lecture est issue de la logique de la figure 6.

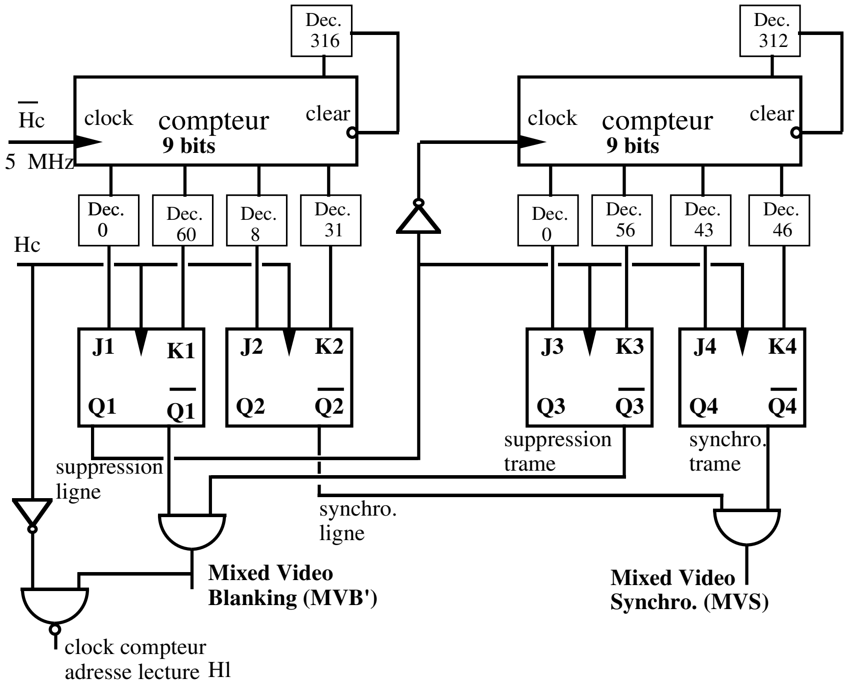


Figure 6: Logique de synchronisation de la vidéo.

Principe de la logique: Pour construire la logique qui produit les chronogrammes désirés, on utilise un compteur de fréquence de base 5 MHz pour le timing ligne (on compte les nombres de coups d'horloge qui définissent chaque séquence de la ligne), et un compteur d'horloge de base de fréquence ligne pour le timing image (on compte le nombre de lignes qui définissent chaque séquence de la trame). Les décodages adéquats des sorties compteurs positionnent des bascules JK en sorties desquelles on retrouve chaque signal de synchronisation et de suppression. Ces signaux seront mélangés aux données digitisées au niveau du DAC de conversion qui ajuste automatiquement les amplitudes des différents signaux vidéo ainsi que l'impédance de sortie.

2.3 Module de mémorisation des données numériques.

La caméra sort des données numériques (codées sur 8 bits) relatives à chaque pixel de la surface photo-sensible, mais aussi relatives à des pixels noirs de référence en début (4) et en fin (3) de ligne, qui ne sont pas intéressants pour la constitution de l'image numérique. Il faut donc isoler les 512 pixels actifs de chacune des 512 lignes par une logique appropriée. Les données sont stockées au rythme de 20 MHz dans une série de 4 mémoires de 128Ko. Normalement il suffirait de disposer de mémoires de capacité 64Ko pour un format 8 bits. Cependant, comme le modèle de la caméra est aussi commercialisé en format 10 bits, on se donne ici la capacité d'un stockage éventuel sur 10 bits en doublant le nombre de mémoires, pour préserver la compatibilité de l'interface à l'évolution du modèle de caméra.

En format pixel de 8 bits, on se donne la possibilité d'optimiser la lecture VME sur 32 bits, en sortant 4 pixels des mémoires par cycle VME de lecture. En format 10 bits, seulement 2 pixels sont lus sur 32 bits par cycle VME de lecture. C'est le bit C5 du registre de commande qui détermine le nombre de pixels à lire. Le module de multiplexage lecture/écriture est à différencier pour les deux cas.

La figure 7 illustre le principe d'écriture/lecture en mémoires numériques.

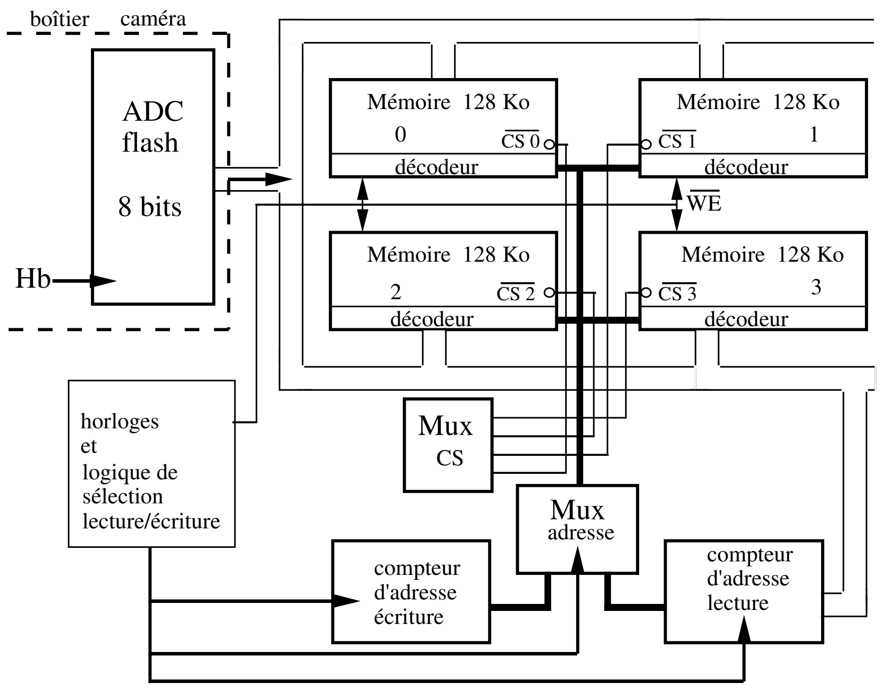


Figure 7: Principe de lecture/écriture en mémoires numériques.

Cas de la lecture à 2 pixels (C5=1):

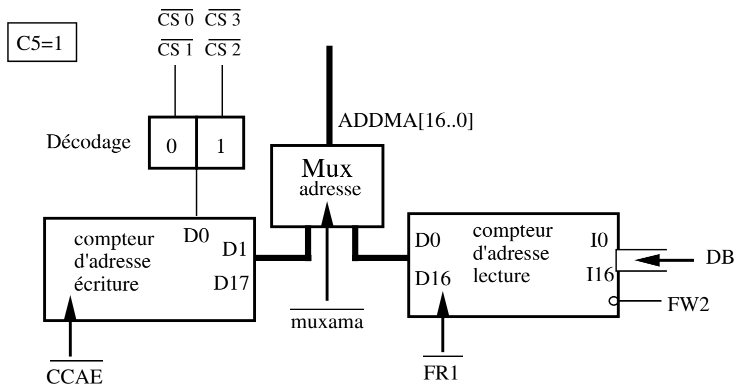


Figure 8: Multiplexage pour la lecture à 2 pixels.

Un premier multiplexeur permet de sélectionner successivement les différents boîtiers mémoires dans lesquels on vient écrire. Ainsi, les pixels pairs sont stockés dans les mémoires 0 et 1. Le premier octet s'inscrit en 0, le deuxième octet (différent de zéro dans le format 10 bits) en 1. Les pixels impairs sont stockés dans les mémoires 2 et 3... L'écriture dans ces mémoires est alternée entre pixels pairs et impairs. Les pixels sont mémorisés un par un, ce qui signifie que le couple de mémoires 0 et 1, ou 2 et 3, est ouvert simultanément pour recevoir les 16 bits d'un pixel pair ou impair. Ainsi, l'aiguillage entre 'Chip Select' pairs (CS0 et CS1) et impairs (CS2 et CS3) se fait en utilisant le bit de poids faible D0 du compteur d'adresse écriture en entrées multiplexeur. A chaque coup d'horloge sur le compteur adresse, on permute donc les niveaux actifs sur les sorties CS.

Cas de la lecture à 4 pixels (C5=0):

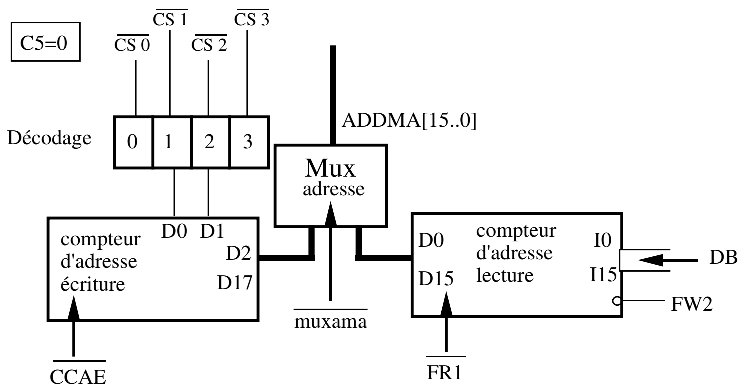


Figure 9: Multiplexage pour la lecture à 4 pixels.

Le multiplexeur permet de sélectionner successivement les quatre boîtiers mémoires dans lesquels on vient écrire. Par exemple, le premier pixel est stocké dans la mémoire 0, le deuxième pixel dans la mémoire 1, le troisième pixel dans la mémoire 2, le quatrième pixel dans la mémoire 3, le cinquième pixel dans la mémoire 0... Les pixels sont mémorisés un par un, ce qui signifie que les mémoires sont ouvertes alternativement pour recevoir les 8 bits d'un pixel donné. Ainsi, l'aiguillage entre les 'Chip Select' indépendants de chaque mémoire (CS0, CS1, CS2, et CS3) se fait en utilisant les bits de poids faibles D0 et D1 du compteur d'adresse écriture en entrées multiplexeur. A chaque coup d'horloge sur le compteur adresse, on décode une sortie CS différente.

Un second multiplexeur positionne successivement l'adresse d'écriture en mémoires sur une phase d'horloge (état bas), puis l'adresse de lecture en mémoire sur l'autre phase de l'horloge (état haut). Dans le cas de la lecture, les 4 mémoires sont ouvertes simultanément, libérant sur le bus de données 32 bits, soit deux pixels de 16 bits (un pixel pair + un pixel impair), soit quatre pixels de 8 bits (un pixel pair + un pixel impair + un pixel pair + un pixel impair). Le CS concerne donc simplement les phases d'écriture pour déterminer le ou les numéros de la mémoire associée à un pixel donné. Il n'y a pas d'aiguillage mémoire particulier pour la lecture. Le compteur lecture peut être chargé en entrées (I0 à I16) ou de (I0 à I15) si on demande la lecture d'une zone particulière de la mémoire. Les données sont lues par accès VME ou VSB depuis le serveur de GalaXie. Le client permet de visualiser l'image numérique avec la résolution maximale de 512x512 points, ou avec des résolutions moindres par regroupement soft des pixels.

2.4 Module de signaux de commande.

2.4.1 Intégration:

Dans la caméra il existe un 'registre programmable' (positionner par deux switchs) qui précise le mode de fonctionnement souhaité:

- pseudo-vidéo.
- synchrone continu (intégration continue soit toutes les 16 ms).
- asynchrone (choix de la fréquence et de la durée d'intégration).

C'est ce dernier mode qui nous intéresse plus particulièrement. Nous distinguerons le mode de fonctionnement asynchrone par défaut piloté par trigger interne (limité à 50 Hz max.), et le mode de déclenchement par trigger externe optimisé pour travailler à des cadences de 60 Hz. D'une manière générale, une impulsion de trigger commande le début de l'intégration des charges sur le capteur. A la fin de cette intégration, les charges sont transférées en zone de stockage avant d'être lues et dirigées vers le module de conversion analogique numérique.

Cas du trigger interne: Une nouvelle impulsion de trigger peut être prise en compte en fin de trame pour redéclencher une nouvelle intégration. La fréquence maximale est calée sur celle de la vidéo soit 50 Hz. La fréquence d'intégration dépend en fait du temps d'intégration. La figure 10 montre un exemple de déclenchement sur trigger interne.

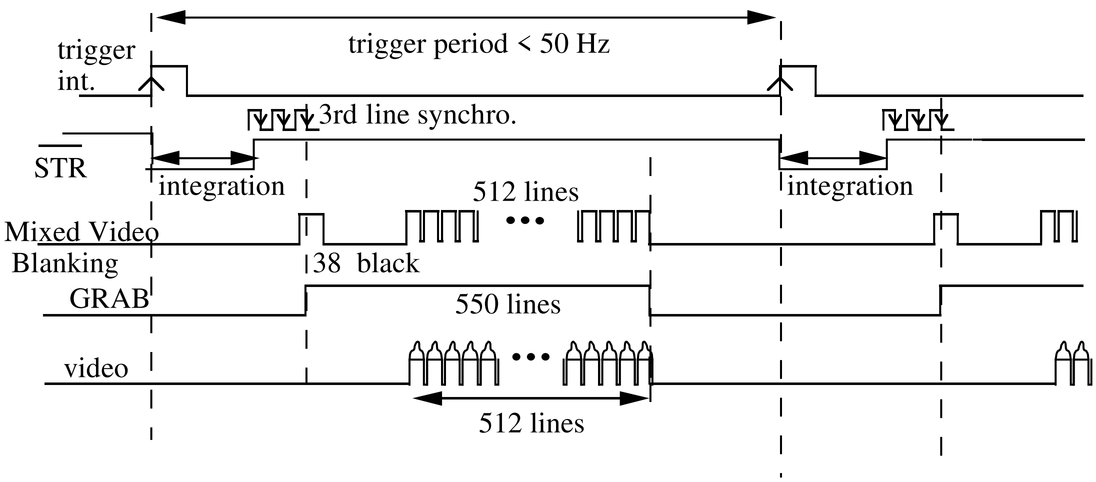


Figure 10: Principe de déclenchement sur trigger interne.

Cas du trigger externe: Pour optimiser la vitesse d'écriture des pixels en mémoires sur l'interface, il est souhaitable de terminer une nouvelle intégration juste après la fin de trame de l'image précédente, de façon à amorcer directement le transfert des charges autour de la descendante du signal GRAB. Il est ainsi possible de rafraîchir les mémoires à environ 60 Hz pour des temps d'intégration compris entre 100 μs et 17 ms (550 fois la durée d'une ligne). Cependant, rien n'empêche de travailler à des fréquences d'intégration plus basses, il suffit d'espacer dans le temps les impulsions de commande, ou d'augmenter les temps d'intégration. La figure 11 montre un exemple de déclenchement sur trigger externe pour des temps d'intégration inférieurs

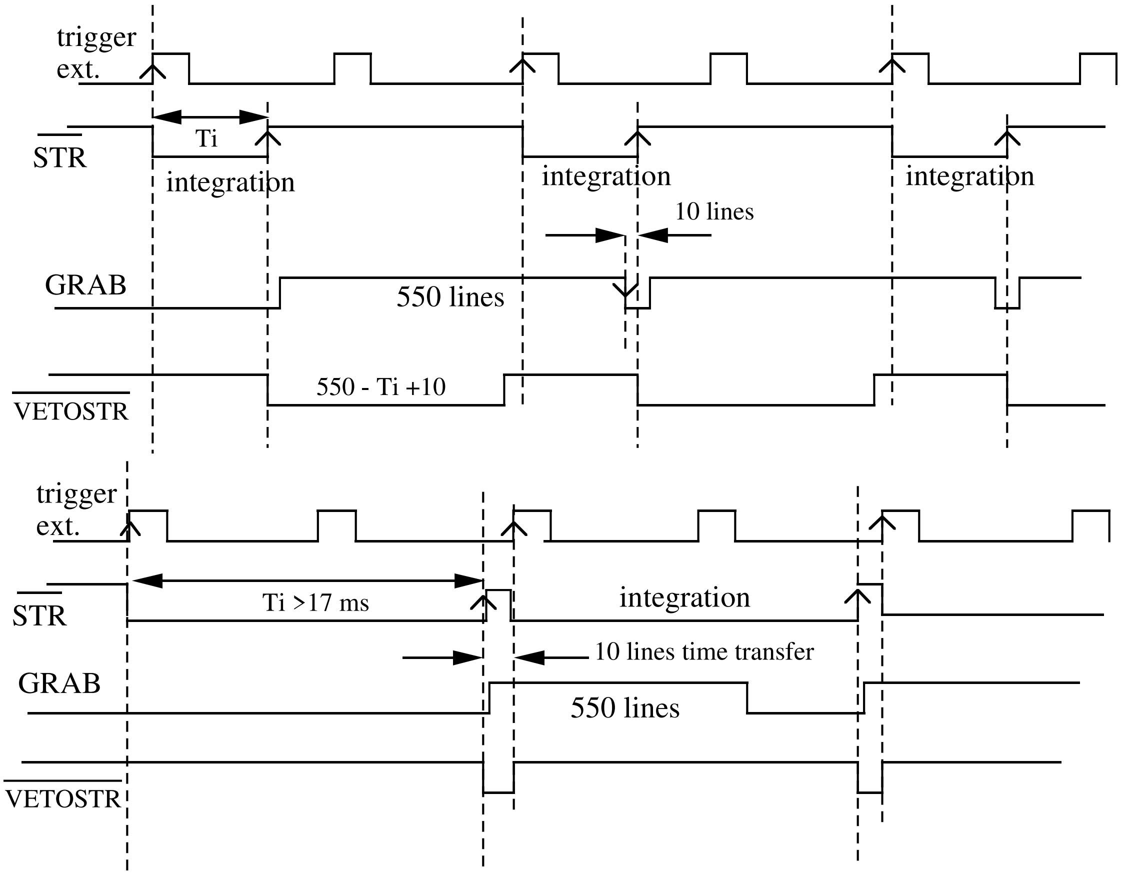


Figure 11: Principe de déclenchement sur trigger externe.

à 17 ms, puis supérieur. Il est à noter qu'il n'y a pas de contraintes particulières sur la durée de ces impulsions de trigger externe, puisque ce n'est pas leur durée qui détermine directement la durée d'intégration. De plus, le fait de commencer une nouvelle intégration avant la fin d'acquisition de l'image précédente en mémoires, ne modifie pas les informations contenues dans celles ci. De ce fait on peut anticiper l'intégration. Les informations de l'image précédente commencent à être surécrites en mémoires juste après le transfert de trame sur le capteur. Il est donc nécessaire d'attendre la descente du GRAB avant le transfert de charges, pour s'assurer d'avoir stocker une image entière en mémoires interface. Dans le cas où Ti est inférieur à la durée des 550 lignes, on impose 10 lignes de sécurité pour s'assurer que le GRAB est bien redescendu avant une nouvelle montée du STR. Dans le cas où Ti est supérieur à la durée des 550 lignes, on impose 10 lignes de sécurité pour s'assurer que les charges sont bien transférées sur le capteur avant une nouvelle intégration (montée du STR). Une logique appropriée permet de caler la remontée du signal STR après la fin de trame GRAB. Pour cela on génère un signal VETOSTR qui permet de ne pas prendre en compte les impulsions de trigger tant que son niveau est à l'état bas (donc pas de nouvelle intégration). Dès que son état passe à '1', le prochain trigger envoyé est pris en compte pour déclencher une intégration anticipée (STR descend à 0) et qui se termine 10 lignes après la fin d'acquisition image. Une logique de comptage permet d'ajuster le temps de VETOSTR à l'état bas en fonction de la durée de temps d'intégration. C'est la fonction FW9 qui charge automatiquement cette durée de comptage dans un registre. Il est nécessaire de préciser le mode de fonctionnement et de passer de l'un à l'autre de façon simple et pratique. Il est donc prévu en face avant un interrupteur qui positionne le hardware

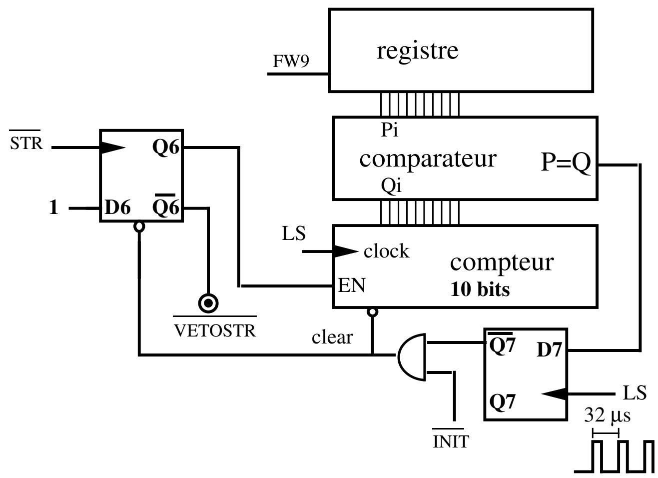


Figure 12: Logique de construction du signal de VETOSTR.

en mode asynchrone sur trigger externe ou interne. Ce même choix sera également possible depuis le client graphique de GalaXie, et sera prioritaire sur le positionnement hardware.

Quelques remarques sur le temps d'intégration:

- *minimum d'intégration:* Si le temps d'intégration devient très court (typiquement inférieur à 1 ms), et donc voisin du temps de décalage ligne à ligne des charges de la zone photosensible vers la zone de stockage (temps de transfert), alors il y a fort risque de bavage vertical ('smearing'). Cet effet est d'autant plus marqué que le faisceau incident est énergétique. En effet, l'intégration se poursuivant pendant tout le transfert, le surplus de charges apporté par un faisceau intense sur les lignes transférées en dernier est alors nettement visible (d'où des traînées verticales possibles).
- *maximum d'intégration:* Il n'y a pas normalement de limite supérieure sur la durée d'intégration. Cependant il faut éviter les excès, pour éviter les risques de saturation.

Construction du signal du temps d'intégration sur déclenchement de l'impulsion de trigger:

- Le temps d'intégration est un multiple entier de $32\mu s$. Par conséquent, on aura

$$3 \times 32\mu s \leq t \leq 32767 \times 32\mu s \simeq 1s$$

Le choix du temps d'intégration souhaité est déterminé par soft depuis le client graphique de GalaXie. Il faut charger $3 \leq n \leq 2^{15}-1$ dans un registre, puis incrémenter un compteur avec une horloge synchronisée sur la synchro. ligne (LS) de période $32\mu s$. Le début du comptage est déclenché par l'impulsion de trigger externe ou interne, avec un jitter de $32\mu s$, car ce trigger et l'horloge LS sont asynchrones. C'est le bit C1 (positionné par soft ou par hard) du registre de status qui détermine le mode de trigger à prendre en compte. Le trigger ne sera autorisé que si le bit C0 du registre de status est à '1', ce qui signifie que la caméra est en mode d'acquisition. De la même façon le trigger ne sera pas autorisé tant que le signal VETOSTR sera à l'état bas (cf. figure 11). Lorsque le compteur atteint n, la sortie comparateur active la remise à zéro des bascules et du compteur. C'est la fin de l'impulsion de temps d'intégration qui a duré $n \times 32\mu s$. La figure 13 reprend le principe de la logique de construction de l'impulsion de temps d'intégration.

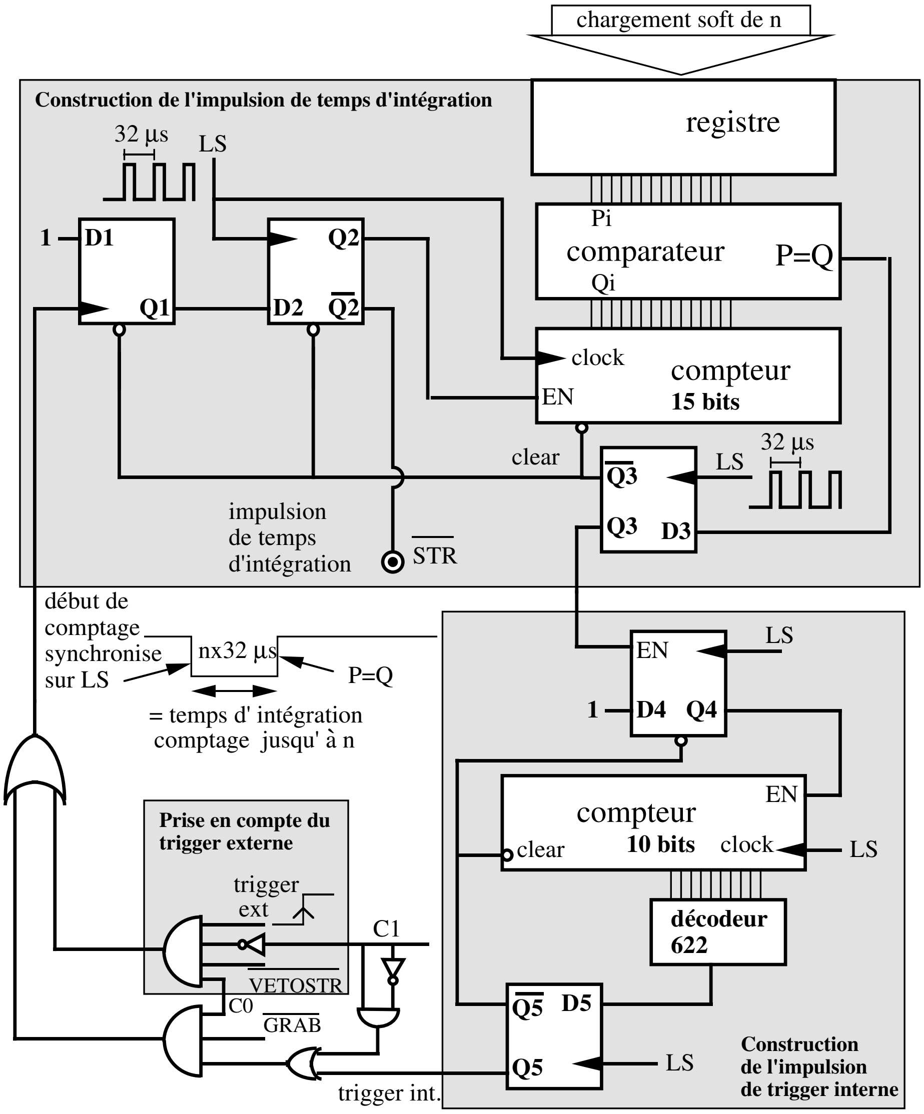


Figure 13: Logique de construction de l'impulsion de temps d'intégration.

Il faut prévoir une configuration par défaut qui sera le minimum de durée d'intégration soit $96\mu s$.

2.4.2 Contrôle de gain:

Le contrôle de gain est une commande en tension qui agit sur les amplificateurs de sorties des registres lecture. En mode de fonctionnement asynchrone, cette commande de gain est programmable par application d'une tension entre 0 et +10V (20 dB), depuis le client graphique de GalaXie. La tension choisie sera convertie par un DAC 12 bits sur l'interface.

2.5 Module de sortie interruption sur numéro de ligne.

Le mode de fonctionnement par interruption sera autorisé ou non, par le contrôle du bit 9 du registre de commande (signal INTEN). Après chaque digitisation d'une ligne et son ac-

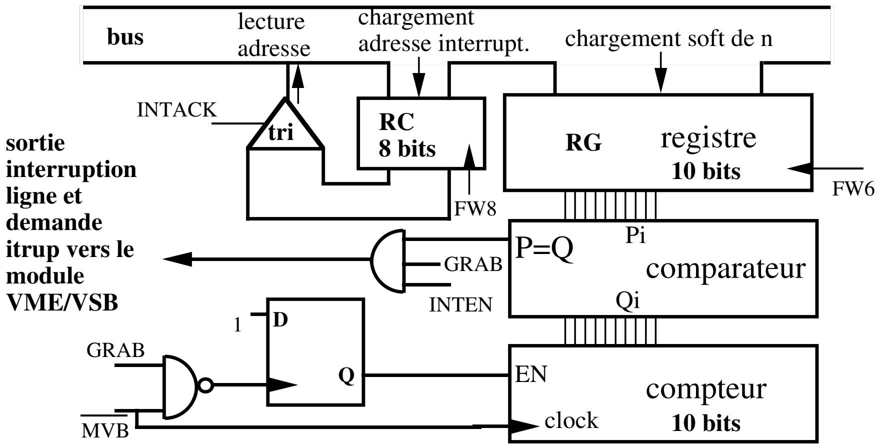


Figure 14: Logique de construction de l'interruption sur numéro de ligne.

quisition complète en mémoires, un compteur de ligne est incrémenté. La valeur du compteur est comparée à celle qui est chargée dans un registre et qui indique le numéro de la ligne oú l' interruption sera générée. Lorsque ce numéro de ligne est atteint par le compteur, une impulsion (de durée égale à la durée d'une ligne) est générée. Elle est dirigée en sortie interruption en face avant de l'interface (donc on pourra en disposer pour synchroniser un DAC de contrôle de bobines miroir par exemple), et vers un module de génération d'interruption qui autorise la lecture de l'adresse de la routine à exécuter (par ex. l'interruption générée sur le bus peut activer une boucle de lecture de données en mémoires et de calculs de signaux d'erreurs pour le contrôle des miroirs). La figure 14 illustre ce principe.

2.6 Registre de commande.

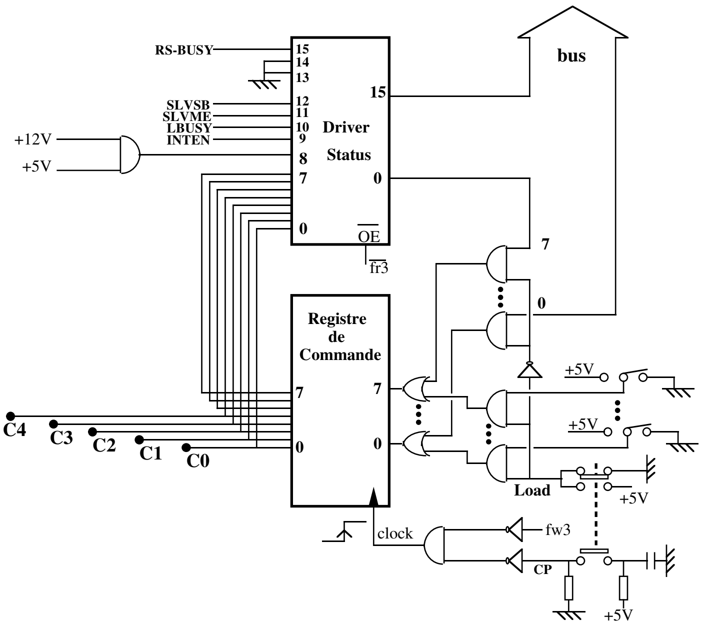


Figure 15: Principe de programmation depuis la face avant du registre de commande.

Ce registre reçoit les différents signaux de commande par programmation soft (chargement des entrées via le bus) ou par positionnement manuel (load des 8 premiers bits) d'interrupteurs en face avant de l'interface. Voici les 8 commandes programmables:

- $C0 = 0:$ Acquisition caméra off.
  $C0 = 1:$ Acquisition caméra on.
- $C1 = 0:$ Trigger externe. (cf 2.4.1)
  $C1 = 1:$ Trigger interne.
- $C2 = 0:$ Lecture VME.
  $C2 = 1:$ Ecriture VME.
- $C3 = 1:$ Reset.
- $C4 = 0:$ Alim. caméra off.
  $C4 = 1:$ Alim. caméra on.
- $C5 = 0:$ Lecture à 4 pixels.
  $C5 = 1:$ Lecture à 2 pixels.
- INTEN $= 0:$ interruptions non autorisées.
  INTEN $= 1:$ interruptions autorisées.
- RS-BUSY $= 1:$ Reset occupation du VME ou VSB.

Cas de C2:
Ce bit précise si l'écriture sur le VME est autorisée ou non. L'écriture dans les mémoires via le bus VME est permise pendant le chargement de données par soft depuis un programme de test par exemple. Dans ce cas il faut interdire l'acquisition caméra (C2=1 C0=0). Dans tous les autres cas (acquisition des données digitisées issue de la caméra et lecture de ces données via le bus VME) C2=0 C0=1. La lecture est autorisée à tous les instants où le système n'est pas en écriture. Cela concerne une des phases d'horloge pendant l'acquisition et tous les autres instants en dehors de la phase d'horloge écriture en mémoires (rebouclage en lecture). Ainsi, si la fréquence d'acquisition n'est pas très élevée (déterminée par la fréquence du trigger externe), la même image est susceptible d'être relue plusieurs fois.

Toutes les sorties du registre de commande (0..7) sont reliées à un driver de status que l'on peut relire par soft pour connaître la configuration. Cette configuration est complétée par un bit de test (8) des tensions d'alimentation (+12V et +5V) sur la caméra. Le bit (9) autorise ou non le mode de fonctionnement par interruptions. Les bits (10..12) donnent l'état des bus. Le bit (15) est positionné à 1 pour libérer le bus.

2.7 Module de décodage VME/VSB.

La structure du bus VME, qui dispose de quatre niveaux d'accès, est construite autour du concept 'maître-esclave': Le maître a le contrôle du bus, tandis que l'esclave après décodage de l'adresse le concernant, répond à une commande envoyée par le maître, constituant ainsi un cycle indivisible lecture/modification/écriture permettant l'utilisation de sémaphores pour se réserver, par exemple, une ressource commune dans un environnement multiprocesseur. Les quatre niveaux d'accès sont: le 'DTB' (Data Transfer Bus), le bus d'arbitration (gestion des accès bus: ici pas d'accès VME et VSB au même instant, la ligne LBUSY donne l'état d'occupation du bus), le bus d'interruption (prise en charge des demandes d'interruptions suivant 7 niveaux de priorités), et le bus utilitaire (horloge, SYSRESET, alimentation...).

Les lignes du DTB contrôlées par un module maître sont illustrées par exemples sur la figure (16):

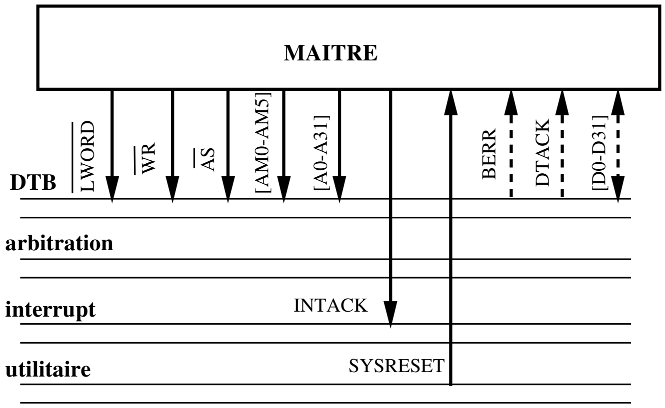


Figure 16: Lignes contrôlées par le module maître.

- 32 lignes d'adresses [A0..A31].
- 6 lignes d'address modifier [AM0..AM5].
- 1 ligne de validation des adresses AS.
- 1 ligne de sélection en mot LWORD.
- 1 ligne de sélection de cycle d'écriture ou de lecture WR.

Les lignes du DTB contrôlées par un module esclave sont illustrées par exemples sur la figure (17):

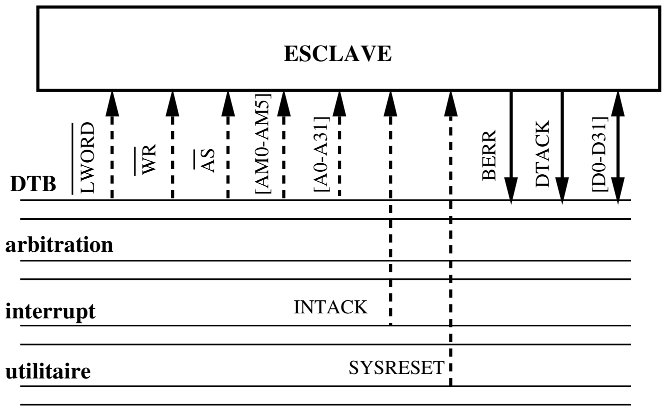


Figure 17: Lignes contrôlées par le module esclave.

- 32 lignes de données [D0..D31].
- 1 ligne d'acquittement de transmission de données DTACK (prise en compte des données lors d'un cycle d'écriture, ou signalement que les données sont stabilisées sur le bus de données en lecture).
- 1 ligne de bus error BERR.

L'adresse VME (P1) est partiellement programmée par soft (20 bits de poids faibles), alors que les 12 bits de poids forts sont positionnés sur la carte par des interrupteurs. Il existe deux formats possibles d'adressage: A24 ou A32.

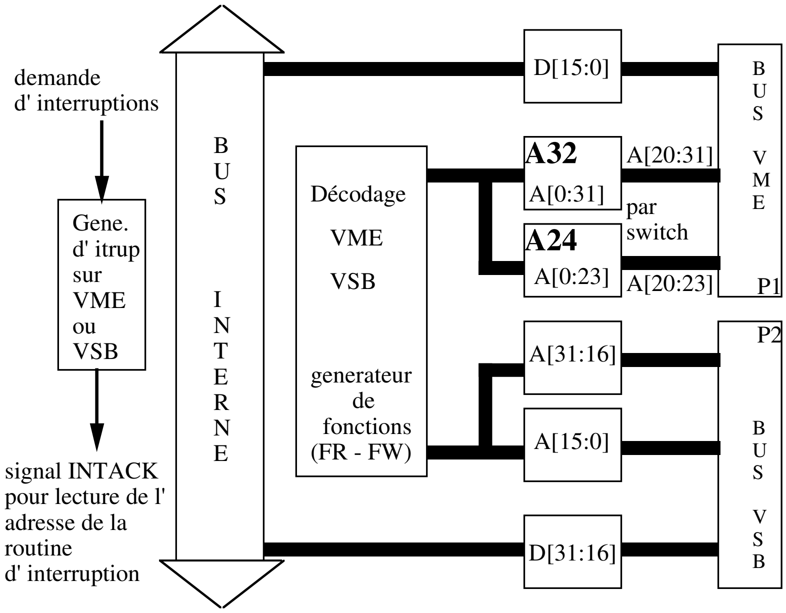


Figure 18: Principe du décodage VME/VSB.

L'adresse VSB (P2) est entièrement programmée en 32 bits. Les adresses sont décodées sur un boîtier de logique programmable qui comporte également la génération des fonctions FR - FW de lecture/écriture:

- FR1: signal de sortie des données mémoires des drivers vers le bus.
- FR2: signal de sortie de la valeur affichée au compteur d'images.
- FR3: signal de lecture du status driver.
- FW1: signal d'écriture de données bus vers mémoires.
- FW2: signal de chargement du compteur d'adresses lecture.
- FW3: signal d'écriture du status dans le registre de commande.
- FW4/FW5: signal de CS et WE dans DAC A / DAC B.
- FW6: signal de chargement d'un numéro de ligne d'interruption.

- FW7: signal de chargement du temps d'intégration dans un registre.
- FW8: signal de chargement du vecteur d'interruption VME ou VSB.
- FW9: signal de chargement de la durée de l'état bas pour veto strobe VETOSTR.

Les données quant à elles, sortent en 2x16 bits. Chaque mot sort sur deux connecteurs différents P1 et P2 pour le VME, et sur le connecteur P2 pour le VSB.

Ce module gère aussi les demandes d'interruptions, notamment celles qui proviennent du module de sortie d'interruption sur numéro de ligne. Le circuit de génération d'interruptions gère la demande en fonction du niveau de priorité (IRQ5 ici, à positionner sur la carte par des cavaliers). Lorsque l'interruption est autorisée, le signal INTACK permet la lecture du vecteur d'adresse (de la routine d'interruption à exécuter) qui sortira sur le bus VME ou VSB pour être dirigé vers le processeur.

2.8 Aspects divers.

- *switch on/off:* Il est important de noter qu'il est interdit de déconnecter le câble qui relie la caméra à l'interface, alors que l'ensemble est sous tension. En cas de nécessité de décâbler la caméra, un interrupteur de mise sous tension on/off (C4 + load) est implanté en face avant de l'interface.
- *relai au mercure:* Il est fortement déconseillé de travailler avec l'interface dans un châssis horizontal, ce qui provoquerait un disfonctionnement voire une dégradation du relai au mercure de l'alimentation.
- *sortie signal frame:* Compte tenu des diverses applications de la caméra, une sortie frame est prévue en face avant. Une impulsion est générée en fin d'image.
- *compteur d'images:* Un compteur 16 bits (dont l'horloge est le signal de frame) sera incrémenté en fin d'image, et sa sortie sera lue par soft depuis le serveur de GalaXie.
- *DAC supplémentaires:* Un autre DAC est mis à disposition pour piloter ultérieurement d'autres signaux de commande (obturateur par exemple).
- *interrupteurs (2 fois 8) sur interface:* pour positionnement de l'adresse (12 poids forts) de la carte. exemple en A32: e8000000 - e80 est la partie haute à positionner par les interrupteurs sur l'interface.

E **8** **0**

1	1	1	0	1	0	0	0
8	7	6	5	4	3	2	1

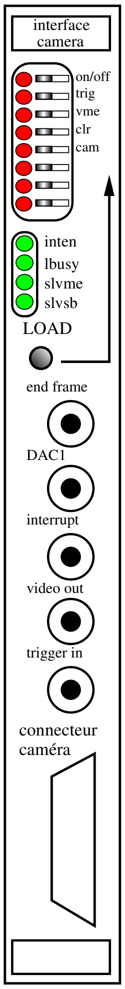
partie haute
(interrupteurs en bas)

0	0	0	0	X	X	X	X
8	7	6	5	4	3	2	1

partie basse
(interrupteurs en haut)

- *cavaliers (S13 à S7) sur interface:* pour niveau d'interruption. IRQ5 (S11) par défaut.
- *cavaliers (S4 à S6) sur interface:* pour address modifier. AM5 par défaut: ! 5 (010).
- *cavaliers (S1 et S2) sur interface:* pour sorties DAC positionnées par défaut.

2.9 Récapitulatif Face Avant.



LED rouges et interrupteurs associés:
on/off: Acquisition pixel (1) ou non (0).
trig: trigger interne (1) ou externe (0).
vme: écriture vme (1) ou lecture (0).
clr: reset (1).
cam: alimentation caméra (1) ou non (0).
+ Bouton LOAD de validation des interrupteurs.
LED vertes:
INTEN: Interruptions autorisées (on) ou non (off).
LBUSY: Carte réservée (on) ou non (off).
SLVME: Bus VME used (on).
SLVSB: Bus VSB used (on).
Connecteurs BNC:
Sortie signal frame.
Sortie DAC A.
Sortie demande d'interruption.
Sortie vidéo.
Entrée trigger externe.
Connecteur femelle D50:
Câble caméra.

3 Simulation des différents modules.

3.1 Regroupement des modules dans des logiques programmables.

Afin de limiter le nombre de circuits sur l'interface, certains modules logiques peuvent être regroupés dans des logiques programmables. Les regroupements sont organisés de façon suivante:

- *Module trigger/intégration:* regroupe toutes les fonctions logiques des figures 12 et 13.
- *Module interruption ligne:* regroupe toutes les fonctions logiques de la figure 14.
- *Module logique vidéo:* regroupe tout le décodage de la mémoire vidéo de la figure 3 (compteurs adresses lecture et écriture, MUX), la logique de sélection lecture/écriture de la figure 4, la logique de synchronisation vidéo de la figure 6.
- *Module de logique d'acquisition:* regroupe toute la génération de signaux d'horloges et de sélection lecture/écriture de la figure 7, le registre de commande et de status driver, et la logique de compteur d'images.

- *Module de multiplexage:* regroupe les multiplexeurs adresse et sélection mémoires (CS), les compteurs d'adresses lecture/écriture des figures 8 et 9.

- *Modules drivers:* regroupe tous les drivers/registres d'entrées/sorties des données sur les bus caméra et VME.

- *Module décodage VME/VSB:* regroupe les fonctions de décodage adresses VME/VSB, et de gestion du bus.

3.2 Simulations.

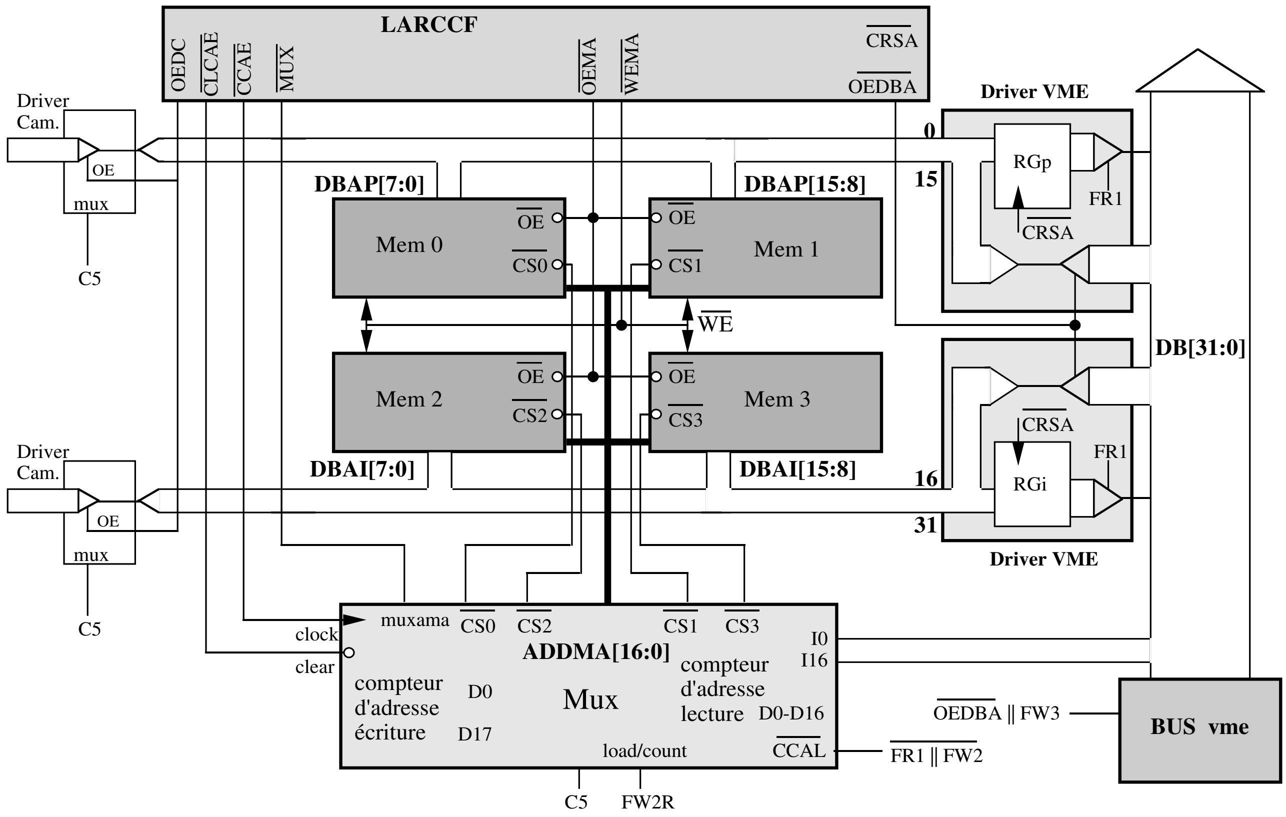


Figure 19: Schéma de simulation de l'acquisition caméra et de l'écriture VME.

Les logiques programmables sélectionnées appartiennent à la famille MAX-7000 PLCC de chez Altera [ref.3]. Cette série de composants offre des circuits avec de grande densité de cellules programmables, et de faibles retards au passage d'une porte (ici 7 ns) ce qui est essentiel pour notre système. Ces circuits sont programmables par le logiciel MAX+PLUS II développé par Altera. Le logiciel permet de compiler, entre autres les fichiers AHDL de description et définitions des signaux d'entrées/sorties et des équations logiques, et de choisir le circuit programmable adéquat ainsi que son brochage pour l'application développée...

La simulation s'exécute suivant un fichier de stimuli Verilog qui décrit parfaitement une séquence de timing d'horloges, de signaux de commande et d'entrées du module à tester. Les timing des signaux de sorties sont fournis par la simulation, ce qui permet de vérifier le bon déroulement de la séquence.

Après avoir effectué les tests de chaque fonctionnalité séparément, les quatre mémoires du module d'acquisition caméra ont aussi été simulées et rattachées à l'ensemble des autres modules pour un test général. Le schéma global de simulation est illustré sur la figure 19. Les résultats obtenus sont illustrés ci dessous.

3.2.1 Séquence d'acquisition en mémoire des données caméra.

Il est très important de vérifier les timings d'acquisition en mémoires des données caméra: Les données caméra entrent par Driver Cam. pour l'écriture en Mem0/1 Mem2/3, et relecture depuis ces mêmes mémoires vers le bus VME via Driver VME.

Cas de la lecture à 4 pixels (C5=0):

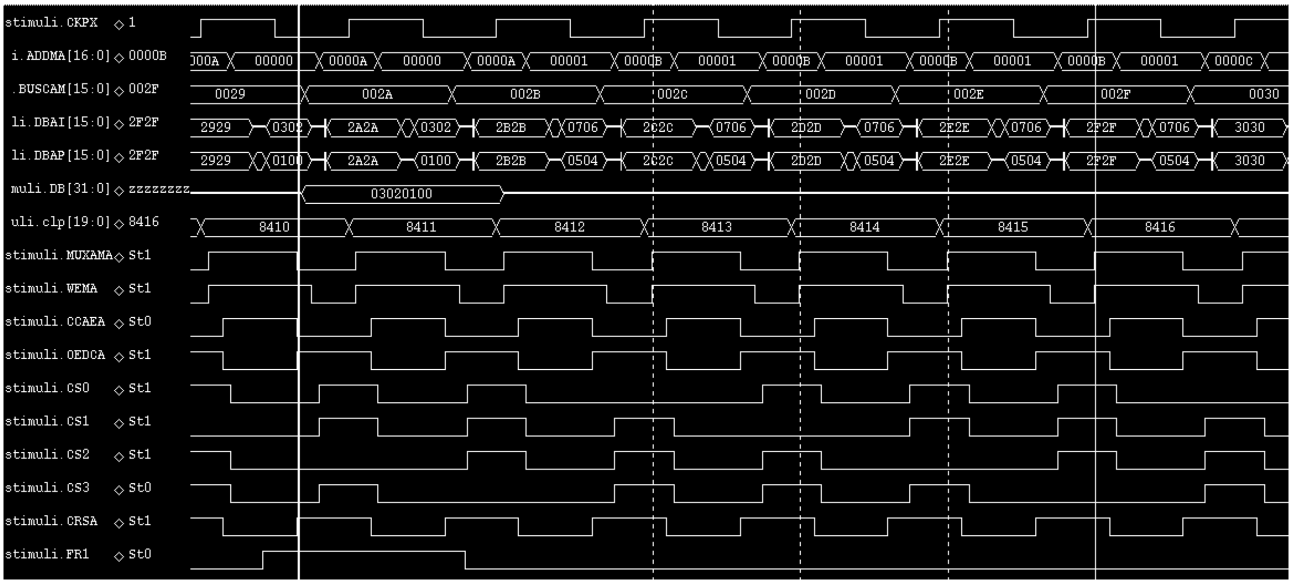


Figure 20: Timing d'acquisition caméra: écriture des données en format 4x8 bits, et lecture 4 pixels via VME.

L'écriture des données caméra en mémoires est cadencée par l'horloge de base de 20 MHz. Pendant une demie période de 25 ns (la demie période suivante est consacrée à la lecture), les données caméra sont présentes (OEDCA = 1) sur les bus mémoires (DBAP et DBAI). Par exemple, lorsque deux pixels 0x2C2C (au deuxième curseur) sont présentés sur ces bus mémoires, alors seule la mémoire paire 0 est sélectionnée (CS0=0 et CS1=CS2=CS3=1) pour l'écriture (validée par la remontée de WEMA). A ce moment, on écrit 0x2C dans la mémoire 0 à l'adresse ADDMA = 0xB. Une période plus tard (troisième curseur), c'est la donnée impaire 0x2D2D qui est présentée sur les bus mémoires. Cette fois ci, on va écrire 0x2D dans la mémoire 1 seulement (CS1=0 et CS0=CS2=CS3=1) à l'adresse 0xB. La période suivante (quatrième curseur), on vient écrire 0x2E dans la mémoire 2 (CS2=0 et CS0=CS1=CS3=1) toujours à l'adresse 0xB. Enfin (cinquième curseur), on vient écrire la donnée impaire 0x2F dans la mémoire 3 (CS3=0 et CS0=CS1=CS2=1). L'adresse écriture ne doit changer que toutes les quatre périodes d'horloge de base. L'écriture suivante concerne la donnée 0x30 que l'on mémorise en 0 (CS0=0) à l'adresse 0xC...

La lecture s'opère éventuellement pendant la deuxième demie période d'horloge. Les bus VME ou VSB ne sont en effet pas capables de transiter des données avec un tel rythme. La lecture est donc beaucoup plus lente. Les drivers VME de sortie des données sont ouverts à une fréquence largement inférieure au 20 MHz. C'est pendant l'état haut du signal FR1 que la lecture est possible. Sur ce niveau, les drivers VME sont chargés sur chaque front montant de CRSA. Sur notre exemple cela correspond à la position indiquée par le premier curseur. On a donc le temps de lire plusieurs fois pendant l'état haut de FR1. On voit que compte tenu du 'débit'des bus VME ou VSB, il ne faut pas changer cette valeur à lire à chaque chargement. L'adresse de lecture ne doit donc pas changer pendant toute une période du signal FR1. A chaque validation de lecture, les quatre mémoires sont ouvertes simultanément (CS0=CS1=CS2=CS3=0) à la même adresse. Sur l'exemple, on lit les deux valeurs 0x00, 0x01 (sur DBAP) et 0x02, 0x03 (sur DBAI) stockées à l'adresse 0x0. Ces valeurs constituent bien une information de 32 bits (en 4 pixels) que l'on retrouve sur le bus DB (qui représente par exemple le bus VME). L'adresse lecture ne doit changer qu'à chaque période de FR1 (sur front descendant).

Cas de la lecture à 2 pixels (C5=1):

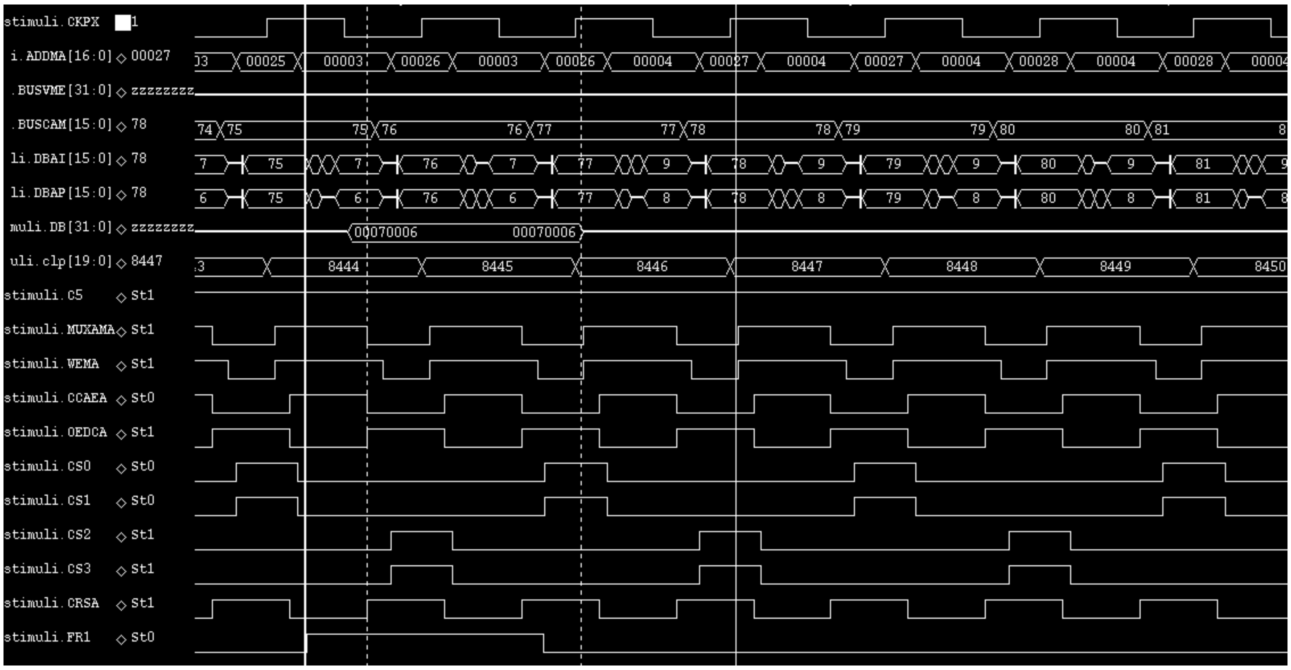


Figure 21: Timing d'acquisition caméra: écriture des données en format 2x16 bits, et lecture 2 pixels via VME.

En faisant basculer C5 à 1, il est possible d'écrire des pixels en mémoires jusque sur une dynamique de 16 bits. On ne peut donc lire les pixels que deux par deux sur le bus VME. Le principe général de lecture/écriture est conservé. Pendant une première demie période de 25 ns, les données caméra sont présentes (OEDCA = 1) sur les bus mémoires pair et impair (DBAP et DBAI). Lorsqu'une donnée impaire est présentée sur ces bus mémoires, par exemple 77 (troisième curseur), alors seules les mémoires 2 et 3 sont sélectionnées (CS2=CS3=0 et CS0=CS1=1) pour l'écriture. Sur ce timing, on écrit 77 dans les mémoires 2 et 3 à l'adresse ADDMA = 26. Une période plus tard, c'est la donnée paire 78 qui est présentée sur les bus mémoires. Cette fois ci, on va écrire 78 dans les mémoires 0 et 1 seulement (CS2=CS3=1 et CS0=CS1=0) à l'adresse 26. La période suivante on viendrait écrire 79 en 2 et 3, mais à l'adresse 27... L'adresse écriture ne doit changer que toutes les deux périodes d'horloge de base.

La lecture s'opère éventuellement pendant la deuxième demie période d'horloge. L'adresse de lecture ne change pas pendant toute une période du signal FR1. A chaque validation de lecture (CRSA), les quatre mémoires sont ouvertes simultanément (CS0=CS1=CS2=CS3=0) à la même adresse. Sur l'exemple, on lit les valeurs paire 6 (sur DBAP) et impaire 7 (sur DBAI) stockées à l'adresse 3. Ces valeurs constituent une information de 32 bits (en deux pixels) que l'on retrouve sur le bus DB.

Le timing suivant montre ce qui se passe à chaque fin de ligne (soit pendant la suppression ligne, où il n'y a pas d'écriture en mémoire).

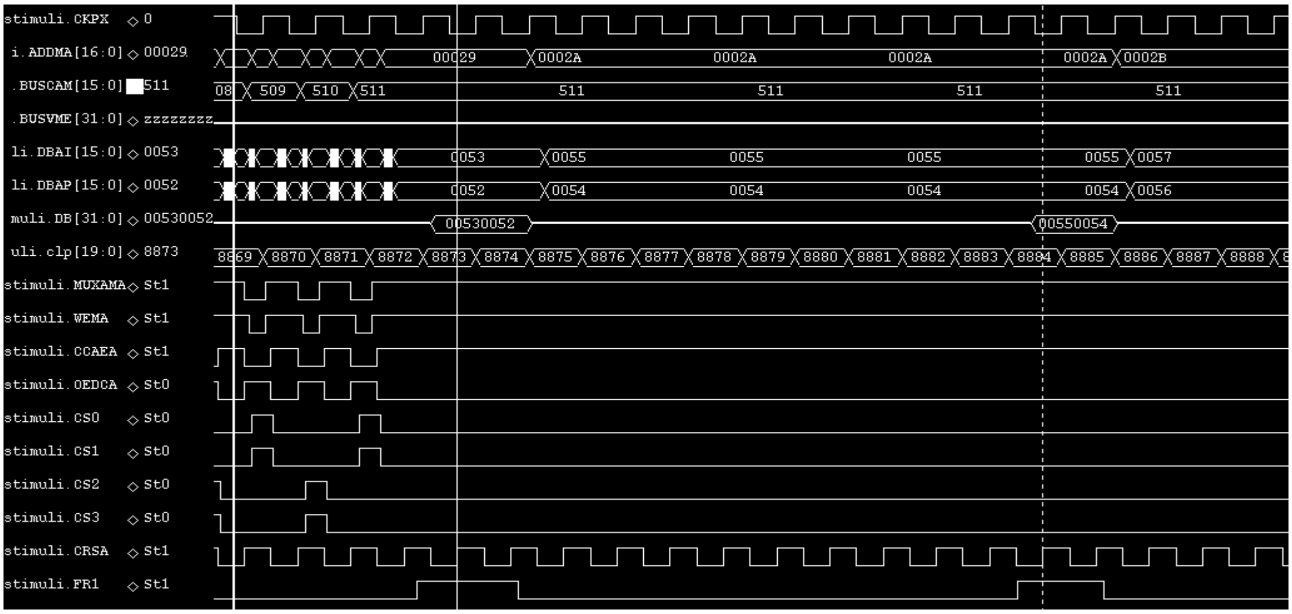


Figure 22: Timing d'acquisition caméra: lecture deux pixels via VME.

La lecture se poursuit toujours au rythme de FR1. La position des curseurs en pointillés indique quelle est la valeur lue. Cette valeur est présente sur le bus DB pendant tout l'état haut du signal FR1 (tri-state ouverts sur drivers), sinon pendant l'état bas du FR1 (tri-state fermés) le bus reprend l'état haute impédance. Cette séquence de lecture seule se poursuit pendant tous les signaux de suppressions ligne et trame, mais aussi entre deux séquences d'intégration image. Si la fréquence d'intégration est faible, il y a rebouclage en lecture (il est alors possible de relire plus d'une fois la même image).

3.2.2 Séquence d'écriture de données depuis le bus VME vers les mémoires.

Cette fois ci, les données arrivent du bus VME pour être écrites en Mem0/1 Mem2/3 via Driver VME.

Pour simuler l'écriture de données en mémoires par un programme soft, on écrit sur les 32 bits de DB, un mot pair (poids faibles) et un mot impair (poids forts) que l'on souhaite mémoriser. Dans notre exemple (C5=1), on veut tout d'abord écrire 0 (DBAP=0) à l'adresse ADDMA = 0 des mémoires 0/1, et 1 (DBAI=1) toujours à l'adresse 0, mais des mémoires 2/3 cette fois. L'écriture paire/impaire ne se fait pas simultanément, tout comme dans le cas de l'acquisition. Dans un premier temps, les mémoires 0/1 sont sélectionnées (SCMP=CS0=CS1=0 et SCMI=CS2=CS3=1) et le front montant du signal WEMA valide l'écriture en mémoires de 0. Lorsque les mémoires 2/3 sont sélectionnées (SCMP=CS0=CS1=1 et SCMI=CS2=CS3=0), on attend le prochain signal de validation WEMA pour écrire 1 dans les mémoires 2/3 toujours à l'adresse 0. Ce n'est que lorsque les deux mots de 16 bits ont été mémorisés que l'on peut changer l'adresse mémoire (un front sur deux de FW1). Donc, après avoir inscrit 0 et 1 à l'adresse 0, au deuxième front descendant de FW1, on cherchera à écrire 2 et 3 à l'adresse 1...

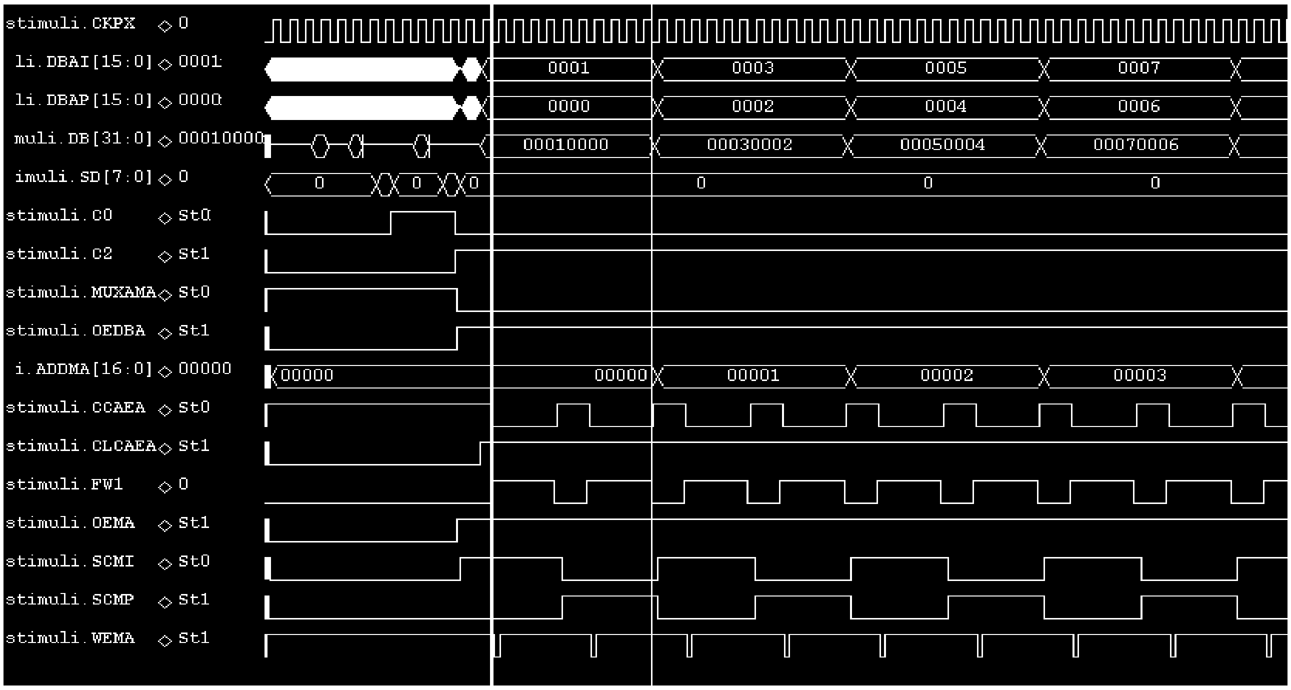


Figure 23: Timing d'écriture en mémoires depuis le VME.

Pendant cette phase C0 est impérativement égal à 0 (pas d'acquisition caméra simultanée), et C2 vaut 1 pour indiquer l'écriture VME.

3.2.3 Construction de la vidéo à partir des données caméra.

Le schéma de simulation est représenté sur la figure 24.

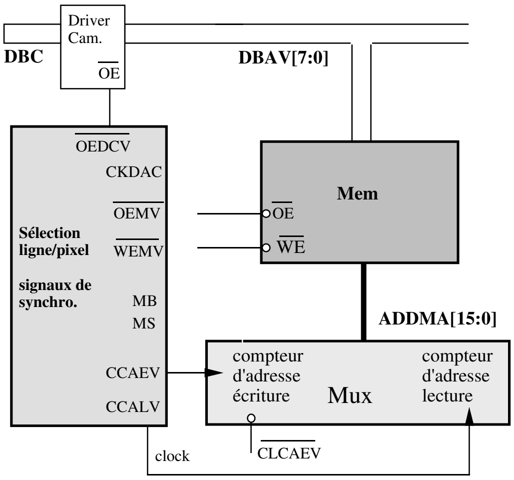


Figure 24: Schéma de simulation de la vidéo.

La simulation concerne la logique de sélection lignes (une sur deux) et de sélection pixels (un sur deux), la logique de construction des signaux de synchronisation vidéo, et d'écriture/lecture en mémoire vidéo. Les données caméra sont transférées sur le bus DBV à travers Driver Cam. Le DAC vidéo de mélange des signaux de synchronisation n'est pas simulé.

Figure 25: Timing global.

Les 512 pixels actifs sortent de la caméra pendant chaque état haut de MVB. L'horloge du compteur écriture (CCAE) est incrémentée pendant ces mêmes instants (une ligne sur deux). Nous verrons sur un timing plus détaillé, la sélection un pixel sur deux. La lecture quand à elle est permanente (horloge du compteur lecture CCAL) sauf aux instants de suppressions lignes (où apparaissent les signaux de synchronisation MS) et trames (non visibles ici, car il faudrait visualiser une image entière).

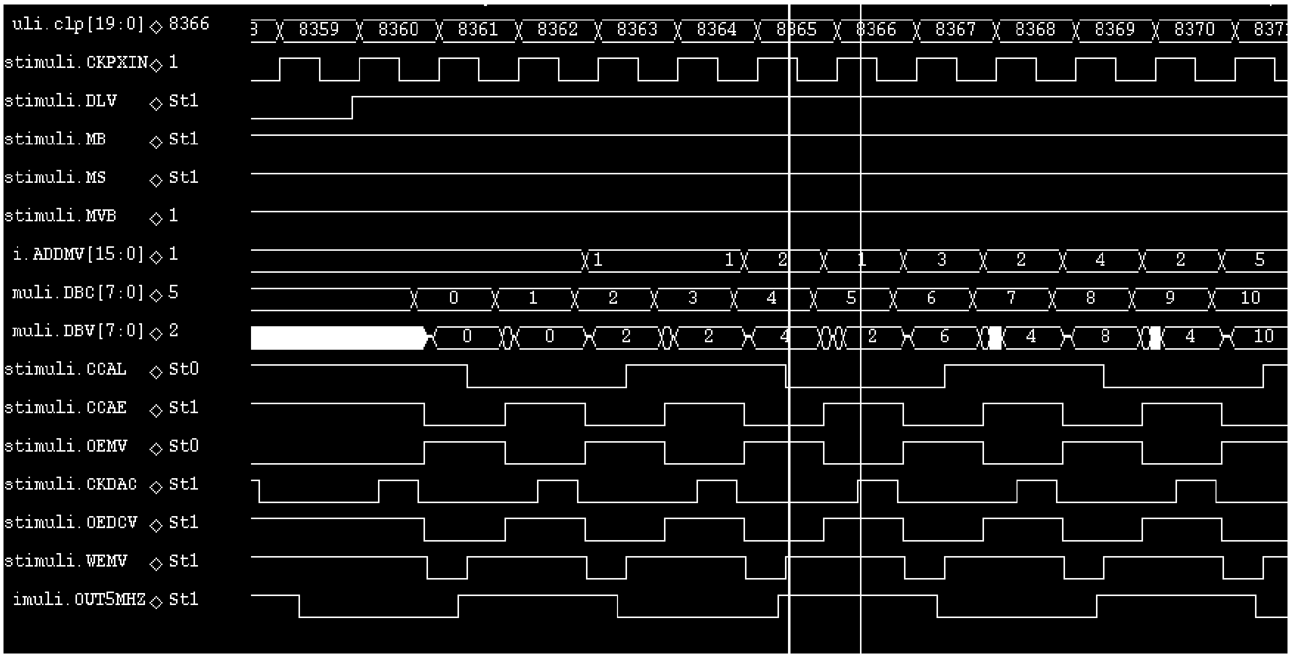


Figure 26: Timing d'écriture/lecture en mémoire vidéo.

Les données sont écrites en mémoire vidéo sur chaque niveau bas de WEMV (et lorsque le driver Cam est ouvert: OEMV=1). Par exemple, au niveau du premier curseur (figure 26), on vient écrire 4 à l'adresse 2 de la mémoire. L'adresse du compteur écriture est ensuite incrémentée (on retrouvera l'adresse 3 sur ADDMV à la prochaine écriture). La lecture est possible quand WEMV et OEDCV sont à l'état haut. Le front montant de l'horloge CKDAC permet de charger la valeur lue dans le DAC vidéo. Par exemple, au niveau du deuxième curseur, on lit le contenu de la mémoire à l'adresse 1. La valeur chargée dans le DAC vidéo est 2. L'adresse compteur lecture est ensuite incrémentée sur front montant de CCAL (on retrouvera l'adresse 2 sur ADDMV à la prochaine lecture). On voit ainsi qu'on sélectionne un pixel sur deux.

3.2.4 Construction de l'impulsion de temps d'intégration.

La figure 27 montre un exemple de déclenchement sur trigger externe pour un temps d'intégration de durées 200 lignes.

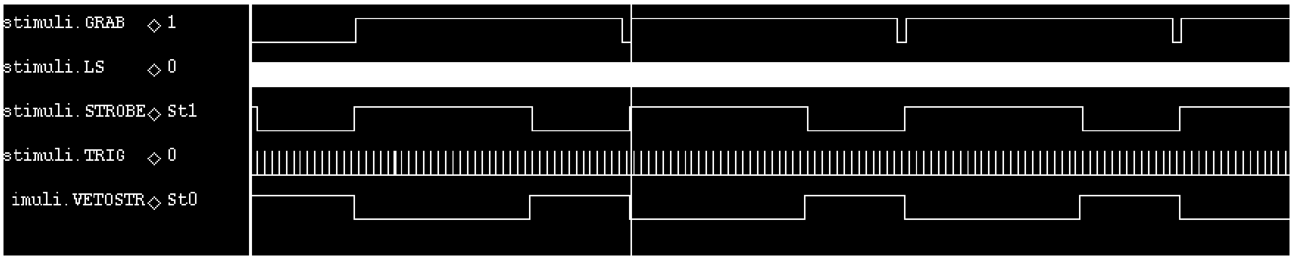


Figure 27: Intégration de durée inférieure à 550 lignes sur trigger externe.

A l'issue de cette intégration, les charges sont transférées sur le capteur (260 μs) puis converties par l'ADC dans la caméra, et stockées en mémoires sur l'interface. Ces opérations se poursuivent pendant tout l'état haut du GRAB (soit 550 lignes). Si on attend la fin de ce signal GRAB pour relancer une nouvelle intégration, alors on perd un temps égal à la durée de l'intégration pendant lequel aucune donnée pixel n'est écrite en mémoires interface. Pour optimiser la vitesse d'écriture des pixels en mémoires, il est alors souhaitable de terminer une nouvelle intégration juste après la fin de trame de l'image précédente, de façon à amorcer directement le transfert des charges directement sur la remontée du STR soit autour de la descendante du signal GRAB. Il est ainsi possible de rafraîchir les mémoires à environ 60 Hz pour des temps d'intégration compris entre 100 μs et 17 ms (550 fois la durée d'une ligne). Dans notre exemple l'état bas de STR (intégration) est égal à 200 lignes. On impose 10 lignes de sécurité pour s'assurer que le GRAB est bien redescendu avant une nouvelle montée du STR. Le sigal de VETOSTR reste donc à l'état bas (pas de prise en compte de trigger) pendant la durée de (550 + 10 - Ti), soit 360 lignes. La fréquence du trigger est ici de 15 lignes.

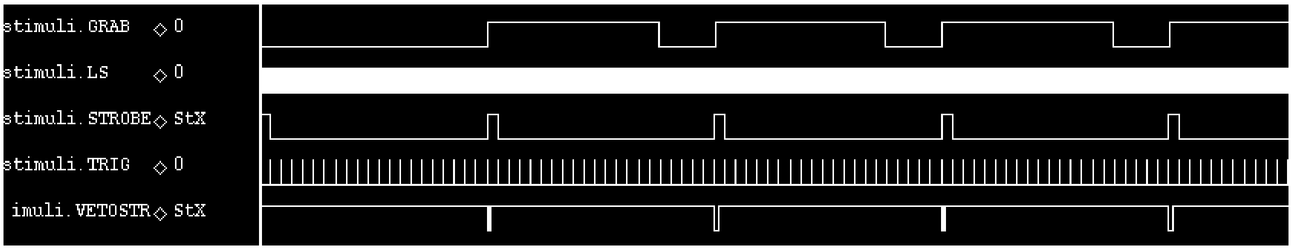


Figure 28: Intégration de durée supérieure à 550 lignes sur trigger externe.

En pratique, on peut utiliser une fréquence optimale de trigger externe voisine de 55 Hz, qui est la durée des 550 lignes, et qui est pratiquement synchrone avec la remontée de VETOSTR.

Dans le second exemple, le temps d'intégration (700 lignes) est supérieur à la durée des 550 lignes. On ne cherche alors plus à caler les remontées du STR sur la descente du GRAB, mais on impose 10 lignes de sécurité pour s'assurer que les charges sont bien transférées sur le capteur avant une nouvelle intégration (montée du STR). La fréquence du trigger est ici de 20 lignes.

4 Tests hardware.

4.1 Mot de contrôle.

Pour configurer la caméra, il faut charger un mot de contrôle dans le registre de commande. Quelques exemples de mots de contrôle sont mentionnés pour les fonctionnalités les plus couramment employées.

Mode acquisition (écriture/lecture)		
lecture 2 pixels	XXXX XX0X XX11 0011	0x 0033 trigger int.
lecture 4 pixels	XXXX XX0X XX01 0011	0x 0013 trigger int.
lecture 2 pixels	XXXX XX0X XX11 0001	0x 0031 trigger ext.
lecture 4 pixels	XXXX XX0X XX01 0001	0x 0011 trigger ext.
camera reset	XXXX XX0X XXX0 1XXX	0x 0008
camera on	XXXX XX0X XXX1 0XX0	0x 0010
écriture mémoires depuis VME	XXXX XX0X XXX0 01X0	0x 0004 ou 0x 0006
lecture mémoires depuis VME	XXXX XX0X XXX0 00X0	0x 0000 ou 0x 0002
interrupt. autorisées + acquisition	XXXX XX1X XXX1 00XX	0x 0231 ou 0x 0233 0x 0211 ou 0x 0213
reset busy	1XXX XXXX XXXX XXXX	0x 8000

4.2 Adresses de base.

En fonction du format choisi, il faut donner une adresse de base de la carte interface, pour définir un espace mémoire de travail. La carte est réservée par positionnement d'un octet 'tas' (0x 80) à l'adresse de base. Les poids forts de la base VME sont positionnés par des interrupteurs sur l'interface. La partie basse est programmée. La base VSB est entièrement programmée. Le tableau suivant donne les intervalles de bases qu'il est possible de choisir en fonction du format d'adressage (cf VmeMapLib.c dans /virgoApp/Vme/pro/src).

format	**base**
VME A24 D32	0x 0. .... $\leq$ base $<$ 0x f. ....
VME A32 D32	0x 00. .... $\leq$ base $<$ 0x fff. ....
VSB A32 D32	0x c00. .... $\leq$ base $<$ 0x cff. ....

4.3 Offsets pour l'adressage des fonctions FR/FW.

Les fonctions de lecture et d'écriture dans les registres et dans les mémoires, sont codées aux offsets indiqués dans le tableau suivant. L'adresse correspond en fait à l'adresse de l'interface auquel on rajoute l'offset.

fonction	offset	
FR1-FW1	0x80000	lect./écriture mémoires - *ctrl_cam
FR2	0x08	lect. compteur d'images - 16 bits - *cpt_cam
FW2	0x08	load compteur d'adresses lect. - 17 bits - *cpt_cam
FR3-FW3	0x04	lecture/écriture status - 16 bits - *ctrl_status
FW4	0x20	load dacA - 12bits - *ctrlA_gain
FW5	0x24	load dacB - 12bits - *ctrlB_gain
FW6	0x1c	numéro ligne interrupt. - 10bits - *cpt_itrup
FW7	0x18	temps d'intégration - 15bits - *ctrl_time
FW8	0x14	adresse d'interrupt. - 8bits - *addr_itrup
FW9	0x0c	load veto strobe - 10bits - *ctrl_veto

Le programme de tests GXtestCamEEV.c est donné en annexe. Les tests comprennent le chargement et la relecture de tous les registres, l'écriture de mots de contrôle et relecture du status, l'écriture de données en mémoires depuis le VME, l'acquisition caméra et la lecture image sur les bus VME et VSB, la lecture image en mode interruption, la lecture image en 'block transfer'... Les librairies (GxLib.c, bltlib.c) pour le fonctionnement en mode interruption et en 'block transfer' sont jointes en annexes. Un driver spécifique à été écrit pour le contrôle de la caméra (GxDrvr.c).

4.4 Résolution théorique.

La précision obtenue sur la mesure de position σ_x, σ_y est limitée par le bruit de l'électronique σ_a, la taille du pixel Δ_x, Δ_y et la dynamique (d) de l'ADC de conversion des charges intégrées sur le capteur en données numériques (niveaux de gris).

Le bruit électronique est mesuré globalement par acquisition d'une image 'noire' dont on a soustrait le pedestal, c'est à dire la lumière ambiante résiduelle. L'histogramme des niveaux de gris des 262144 pixels image donne le r.m.s de la distribution gaussienne associée, soit environ $\sigma_a = 0.5$ coup d'ADC.

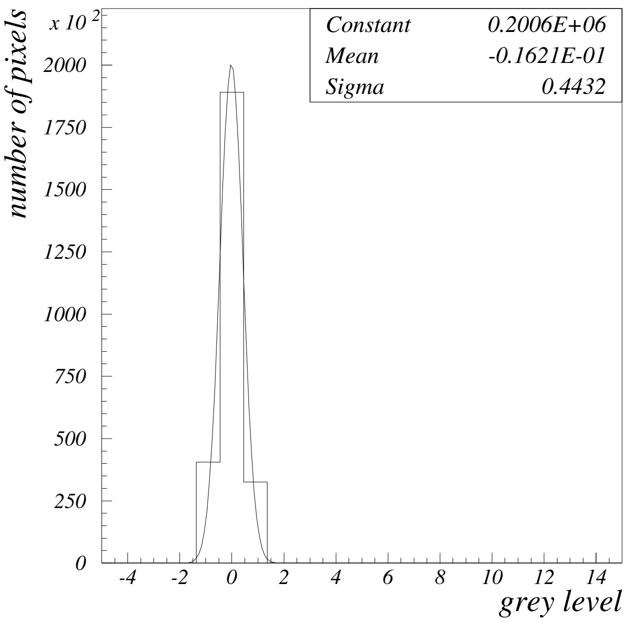


Figure 29: Distribution des niveaux de gris d'une image 'noire' dont on a soustrait le pedestal.

Avec un ADC 8 bits et des pixels carrés de $15\mu m$ x $15\mu m$, la résolution théorique est définie par [ref.4]:

$$\sigma_i \simeq \frac{\sigma_a \Delta_i}{d} \tag{1}$$

et donne $\sigma_x = \sigma_y \simeq 30nm$ par image. Comme la caméra produit un maximum de 60 images par seconde, on peut encore écrire que la résolution maximale est $\sigma_x = \sigma_y \simeq 4nm/\sqrt{Hz}$.

Remarques:

- Le bruit dépend du temps d'intégration (génération thermique de porteurs), pour un gain donné. Pour un temps d'intégration maximum de 1 s, $\sigma > 5$.
- Le r.m.s de la distribution s'élargit si le gain appliqué augmente. Pour un gain maximum de 20 dB, $\sigma \simeq 4$.

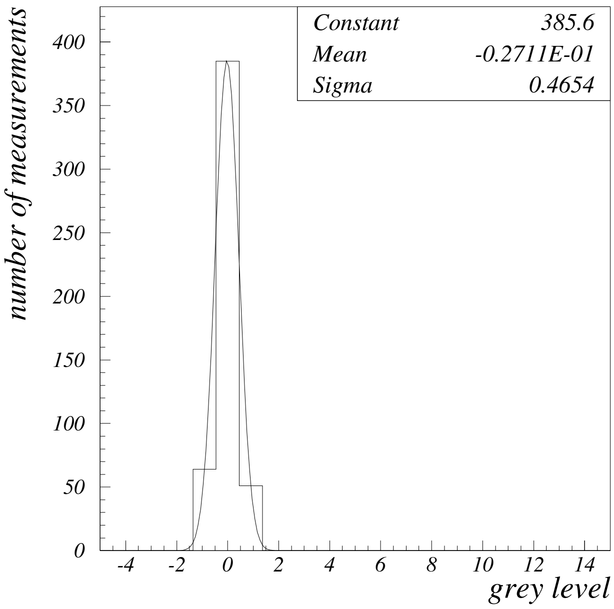


Figure 30: Distribution des niveaux de gris sur 500 images, d'un pixel illuminé.

Il est possible de pousser plus loin les tests, afin de vérifier les fluctuations de niveaux de gris sur un pixel donné à éclairage constant (pour un temps d'intégration fixé). La mesure consite à relever les niveaux de gris d'un pixel, par exemple sur 500 images intégrées successivement en lumière du jour. L'histogramme des niveaux de gris donne le r.m.s de la distribution gaussienne associée, soit environ $\sigma_a = 0.5$ coup d'ADC. On retrouve des résultats comparables à ce que l'on obtient dans le noir pour l'ensemble des pixels.

Il est également intéressant de vérifier comment varie la réponse d'un pixel en fonction du temps d'intégration. Le test consiste donc à mesurer le niveau de gris d'un pixel soumis à des durées variables d'éclairement. Les résultats sont illustrés sur les figures (31). Le premier graphe montre que la variation est parfaitement linéaire pour un pixel éclairé par un faisceau lumineux qui sature rapidement le senseur. Le deuxième graphe illustre la réponse d'un pixel soumis à éclairage ambiant et donc à des temps d'intégration beaucoup plus importants. La variation n'est plus vraiment linéaire entre les niveaux de seuils minumum et maximum. Ici, le niveau maximum est atteint avant saturation, et l'exposition supplémentaire à la lumière ne modifie pratiquement plus ce résultat.

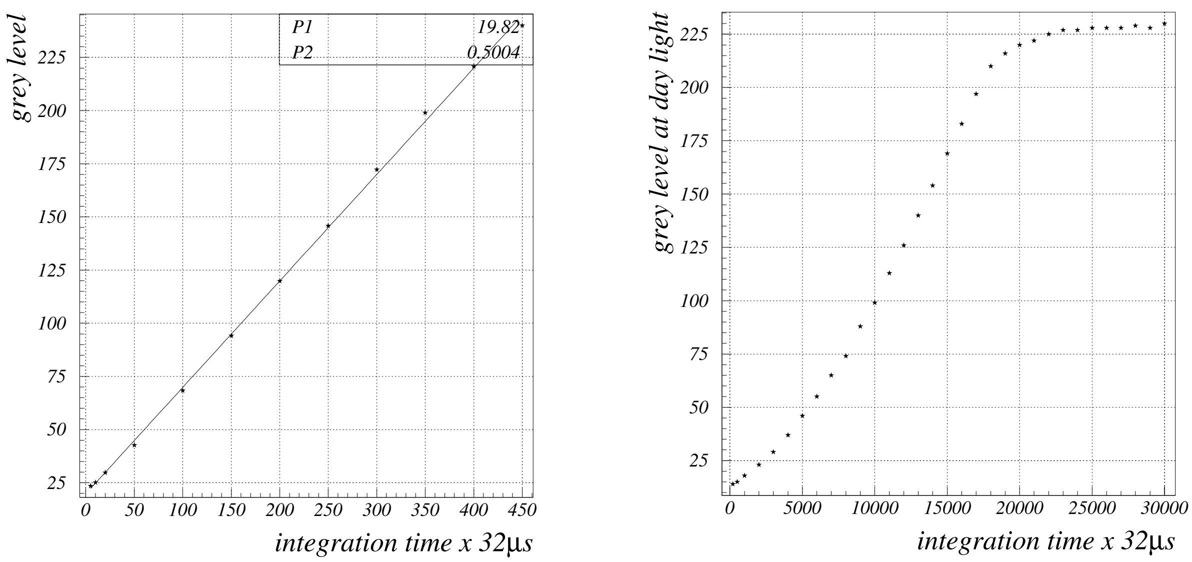


Figure 31: Variation du niveau de gris d'un pixel en fonction de la durée d'intégration.

4.5 Mesure de la résolution.

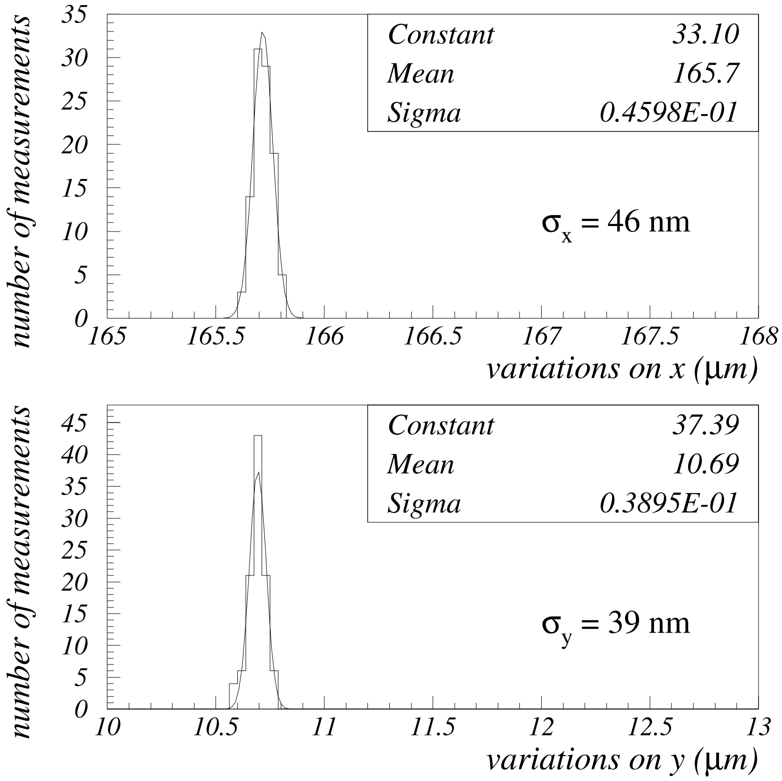


Figure 32: Variations de positions suivant x et y (gain=0).

Pour évaluer la précision sur la mesure d'une position, on crée un point d'impact lumineux sur la surface sensible de la caméra à travers un diaphragme de $500 \mu m$. Le programme calcule en permanence (ici à 20 Hz par exemple) les coordonnées du spot lumineux, et inscrit les résultats dans un fichier. Sur un échantillon de 100 points, on peut représenter les histogrammes des variations de positions suivant x et y.

On retrouve des résolutions inférieures à 50 nm pour des fluctuations rapides de positions. Ces résultats sont compatibles avec la précision théorique de 30 nm.

Un calcul de fft suffit pour déduire des points de mesures, la densité spectrale de positions suivant les directions x et y. Les résultats sont illustrés sur la figure 33.

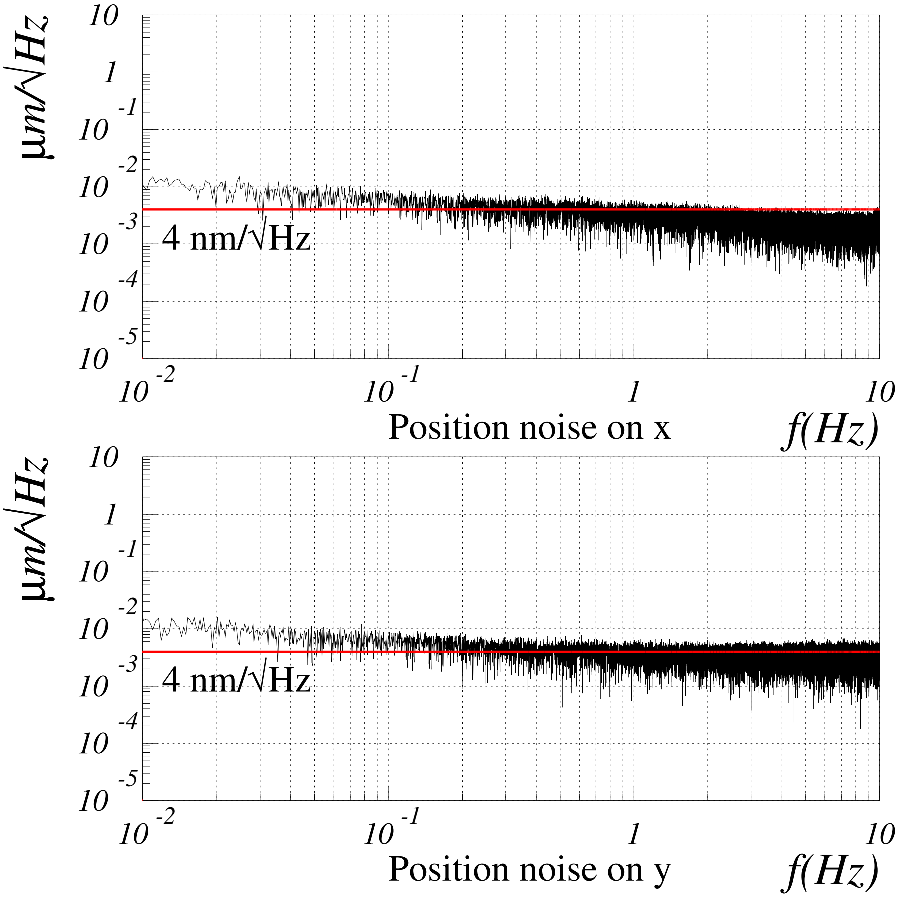


Figure 33: Bruit de mesure de position suivant x et y.

La résolution mesurée est compatible avec la valeur théorique.

4.6 Vitesse de lecture à deux pixels.

Nous avons comparé les vitesses de lecture VME d'une image complète (512x512) pixels, par block transfer et par lecture mot à mot de chacun des pixels image stockés sur les mémoires de l'interface caméra. Rappelons que deux pixels sont lus par mot de 32 bits. Une image correspond donc à 128 Ko. Les temps de lecture sont indiqués ci dessous:

lecture BLT	lecture mot à mot	processeur
30 ms	150 ms	FIC 8234
36 ms	180 ms	RIO 806X

On a donc au moins un gain de temps d'un facteur 5 pour la lecture en mode block transfer.

5 Adaptations soft de GalaXie.

Un unique programme doit fonctionner aussi bien pour l'ancienne caméra (EGG), que pour la nouvelle (EEV). Ceci implique des modifications au niveau du serveur et du client de GalaXie.

5.1 Adaptations du serveur.

Le serveur reconnaît automatiquement le type de caméra pour lequel il doit être exécuté. La reconnaissance automatique est possible grâce au test de positionnement d'un octet (t.a.s) à l'adresse de base de l'interface caméra. Le positionnement n'est pas possible pour l'ancien modèle de caméra. Lorsqu'un client demande la connexion au serveur, celui-ci lui renvoie un message de status précisant quel est le type de caméra utilisée. Le client sait ainsi adapter automatiquement sa configuration de travail pour l'ancien ou le nouveau modèle.

Le serveur est exécuté en acquisition VME ou VSB. Il n'y a pas de possibilités de passage de l'un à l'autre pendant le run.

Les fonctions de lectures des mémoires sur l'interface sont identiques, puisque la logique d'acquisition est la même pour les deux modèles de caméras.

5.2 Adaptations du client graphique.

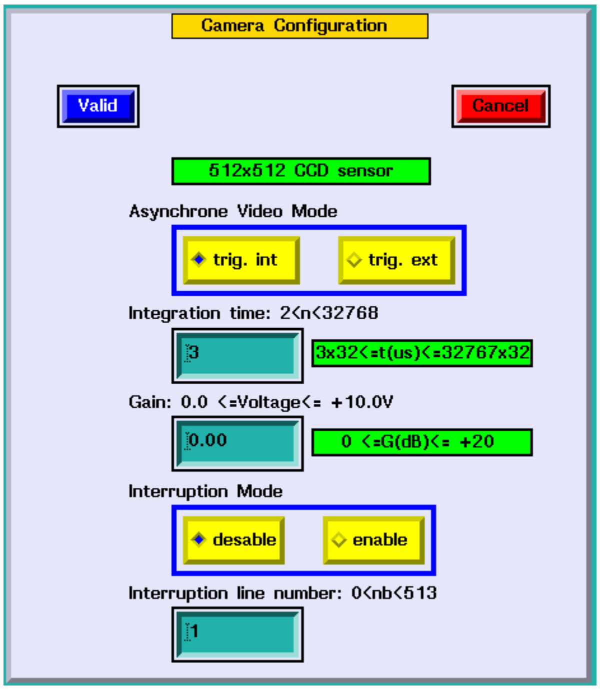


Figure 34: Module de configuration de la nouvelle caméra.

Ce panneau de configuration permet de définir le mode de fonctionnement de la caméra. L'utilisateur peut fixer: le mode de déclenchement de l'intégration image par trigger interne ou externe, la durée du temps d'intégration entre $100\mu s$ et $1s$, le gain entre 0 et 20 dB, le masquage ainsi que le numéro de ligne de déclenchement des interruptions. Les choix doivent être validés. Cette action permet l'envoi de la nouvelle configuration de travail au serveur qui applique les changements hardwares. Les illustrations suivantes représentent des images 512x512 intégrées sur la caméra EEV.

References.

[ref.1] EEV Imaging, Sensing and Displays Division - 'Super photon CCD camera CAM17-16/X219 user guide' - sept(1997)

[ref.2] 'Traité d'électronique' - Techniques de l'ingénieur - E64000 - p7/11.

[ref.3] Altera Corporation - 'MAX+PLUS II Programmable logic development system and software' - v7.22 (1997)

[ref.4] A. Bazan et al. - 'A VME based imaging system for the Virgo project' - Astroparticle Physics 2(1994) 229,234

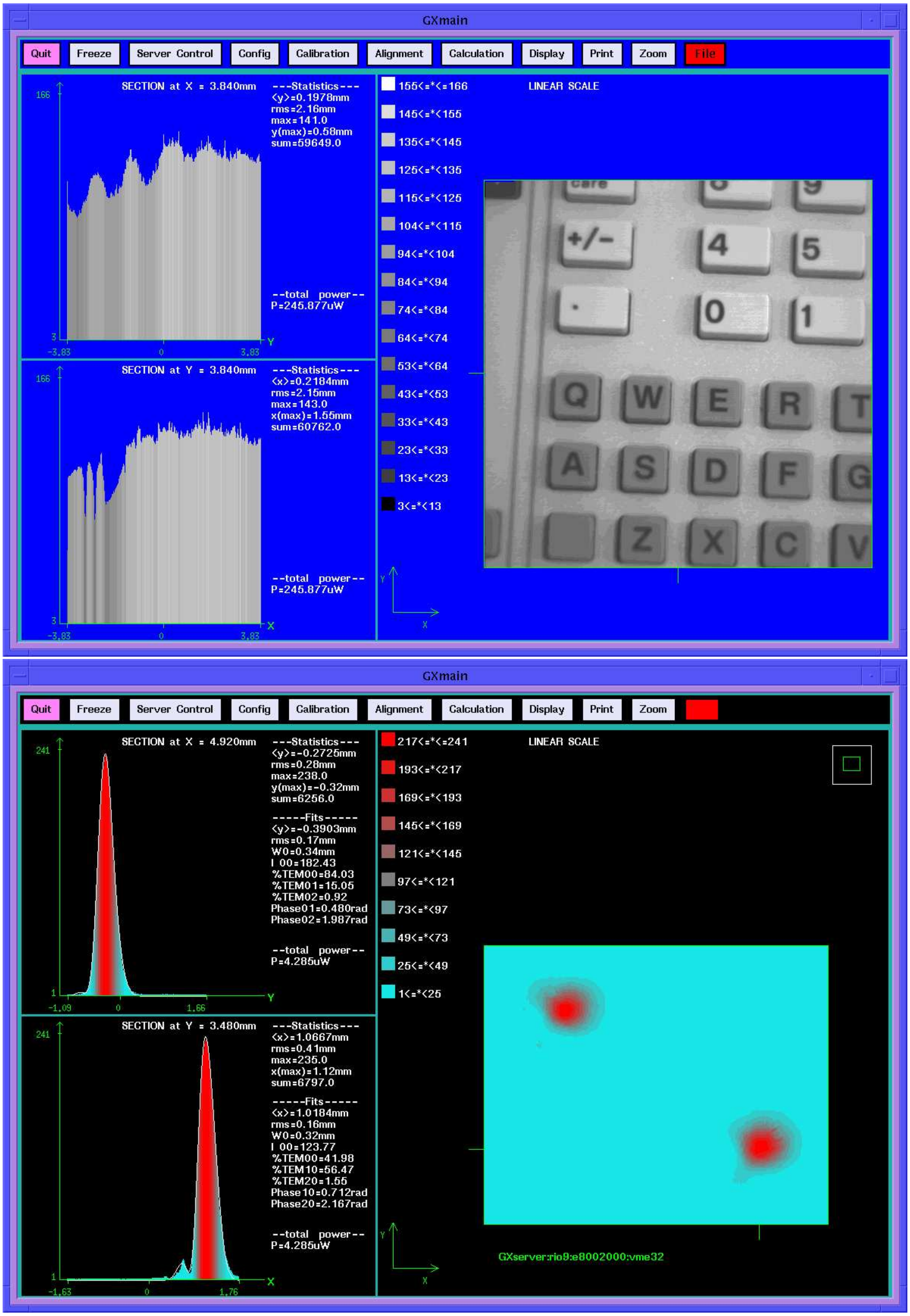
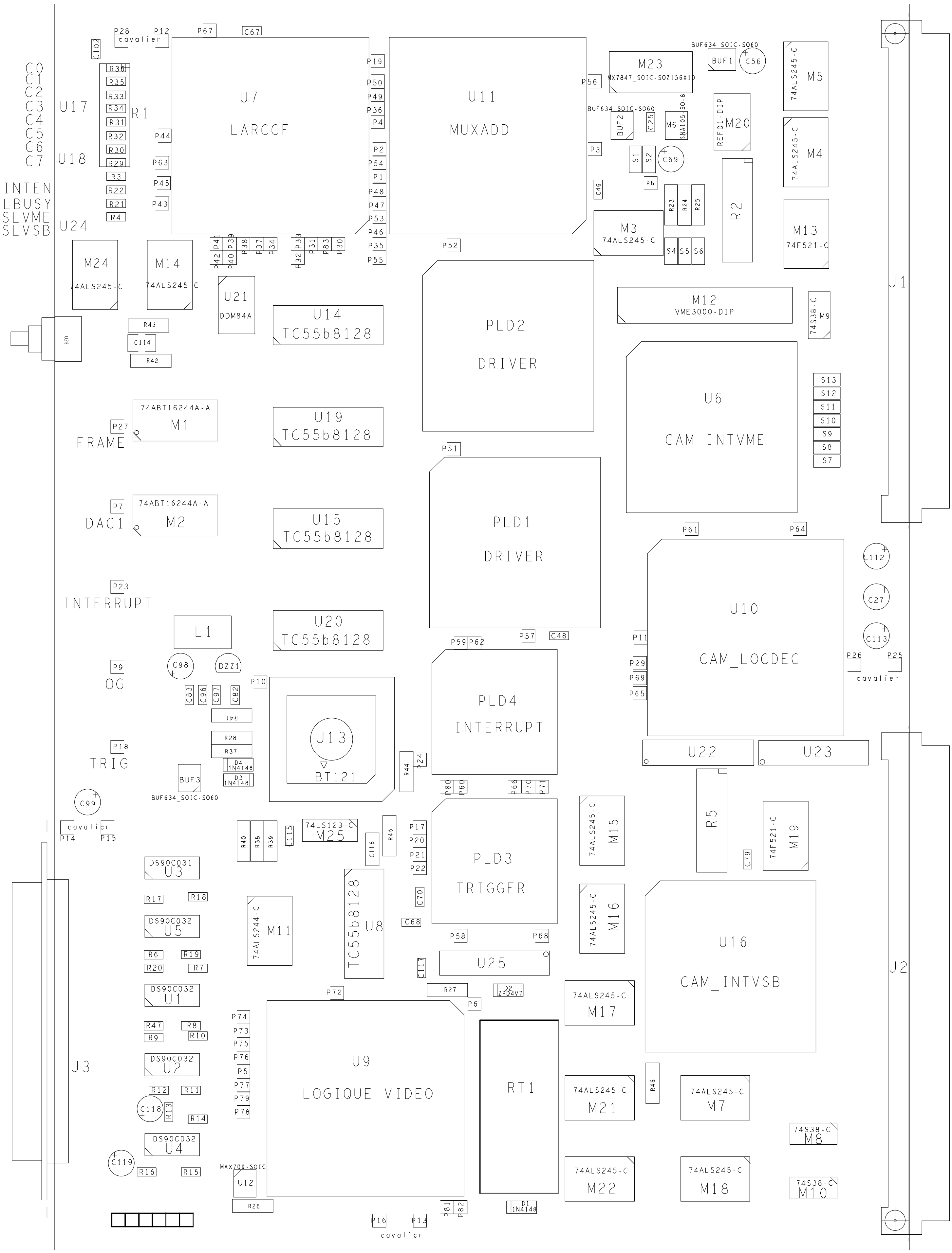


Figure 35: Images 512x512.



C0
C1
C2
C3
C4
C5
C6
C7
U17
U18
R1
INTEN
LBUSY
SLVME
SLVSB
U24
U7
LARCCF
U11
MUXADD
M23
MX7847_SOIC-SOZ156X10
BUF634_SOIC-SO60
BUF1
C56
BUF2
M6
REF01-DIP
M20
C69
R2
M5
74ALS245-C
M4
M13
74F521-C
M3
M24
M14
U21
DDM84A
U14
TC55b8128
PLD2
DRIVER
M12
VME3000-DIP
74S38-C
M9
U6
CAM_INTVME
74ABT16244A-A
M1
FRAME
U19
DAC1
M2
U15
PLD1
U10
CAM_LOCDEC
INTERRUPT
U20
L1
C98
DZZ1
OG
PLD4
U13
BT121
TRIG
BUF3
C99
cavalier
U22
U23
R5
M15
M19
74F521-C
74LS123-C
M25
PLD3
TRIGGER
DS90C031
U3
DS90C032
U5
74ALS244-C
M11
TC55b8128
U8
M16
U16
CAM_INTVSB
U25
U1
M17
U2
J3
U9
LOGIQUE VIDEO
RT1
M21
M7
C118
74S38-C
M8
U4
C119
M22
M18
M10
MAX709-SOIC
U12
J1
J2