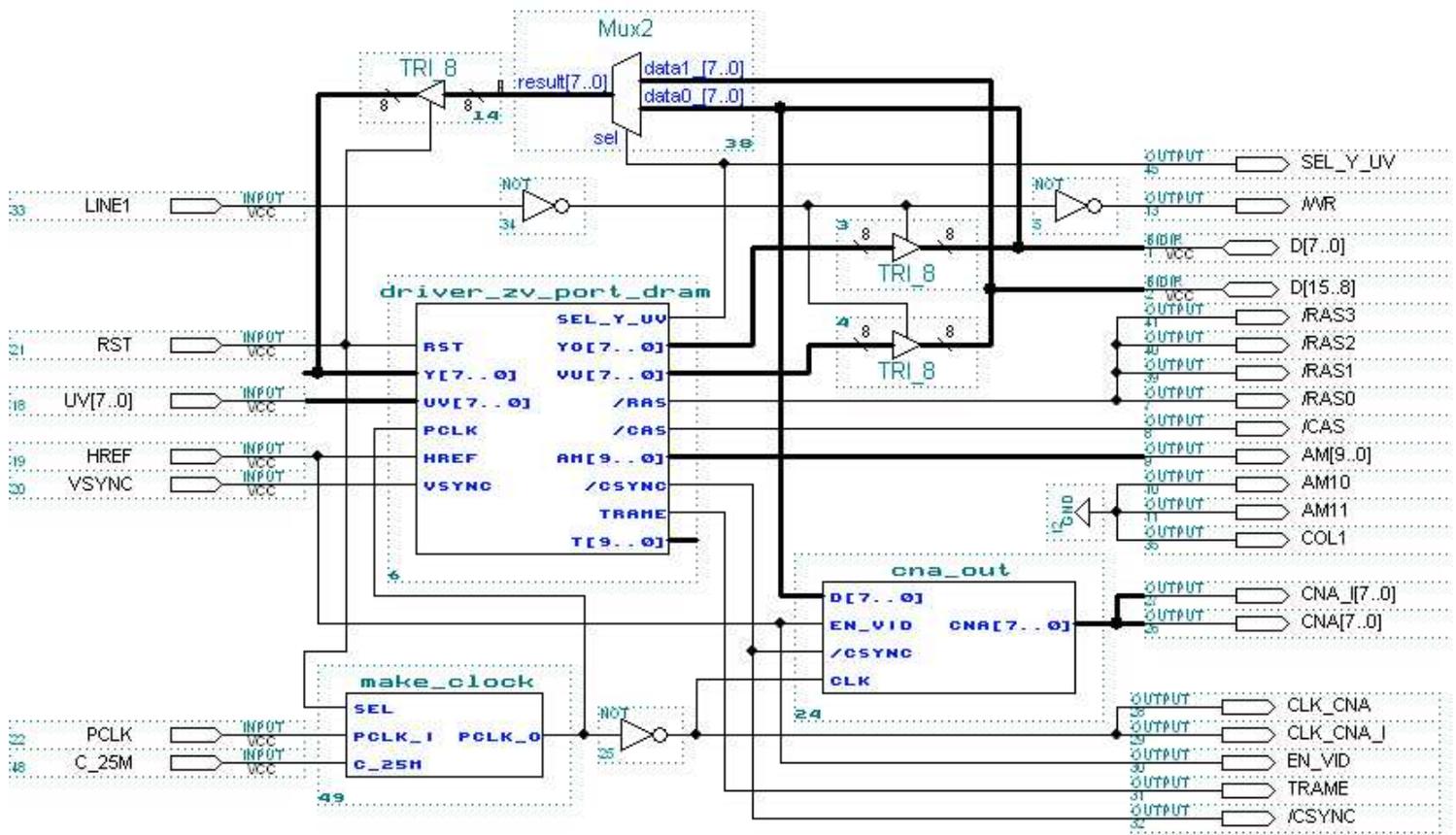


GESTION ZV-PORT & MÉMOIRE D'IMAGE SUR MODULE CAMÉRA C38A (OV7620)

Note : l'image mémorisée peut être visualisée sur un écran N&B connecté sur la maquette d'étude, mais le module caméra est conçu pour permettre une lecture de la mémoire d'image (en couleurs) depuis un PC via son port parallèle.

SCHÉMAS ET DESCRIPTIONS AHDL

1. Schéma principal



Liaisons avec les composants extérieurs à l'EPLD :

- Module caméra C38A : ZV-port (Y[7..0], UV[7..0], PCLK, HREF et VSYNC)
L'état du signal RST est déterminé par le PC connecté au module caméra.
Connexions via connecteur "I_O" 24 broches de la maquette d'étude
- Module RAM dynamique SIMM : D[15..0], AM[11..0], /RAS[3..0], /CAS et /WR
Le module SIMM inséré dans le connecteur RAM doit être compatible EDO.
- Convertisseur N/A 8 bits (production vidéo composite N&B) : CNA[7..0] et CLK_CNA
- Clavier décimal : COL1 et LINE1. LINE1 est à "1" au repos et passe à "0" à l'action des touches 1, 2 ou 3.

2. Descriptions des signaux :

2.1 ZV-port

Le bus "ZV port" est un standard développé par INTEL pour connecter des sources vidéo numériques au port PCMCIA des PC portables. Tous les ports PCMCIA n'accepte pas le bus ZV-port.

Il est constitué des signaux suivants :

- Y[7..0] : luminance codée en binaire naturel. Débit: 13,5M échantillons/s. Étendue : 16 à 252
- UV[7..0] : alternance des composantes de différence couleur CR[7..0] (U) et CB[7..0] (V) codés en binaire décalé. Débit: 6,75M échantillons/s pour chaque composante.
Étendue : -120 à +120
- PCLK : horloge "pixel" de fréquence 13,5MHz. Le flanc montant identifie l'instant d'échantillonnage de Y[7..0] et UV[7..0].
- HREF : validation de l'horloge PCLK pour identifier les pixels "utiles" (ici 640 pixel par lignes sur 480 lignes).
- VSYNC : synchronisation verticale. Passe à "1" pendant qq lignes au début de chaque trame. Fréquence 60Hz (norme NTSC).
Si la caméra est inactive (RST="1"), VSYNC est produit par le PC et utilisé pour initialiser les structures liées à la lecture de la DRAM.

2.2 Pilotage RAM dynamique

On utilise un module DRAM compatible EDO de type SIMM pour mémoriser l'image numérique. La RAM interne à l'EPLD est insuffisante.

- LINE1 : "1" → lecture et "0" → écriture. L'état de repos de LINE1 est "1" et est mis à "0" en appuyant sur une des touches 1, 2 ou 3 du clavier de la maquette.
- RST : reset du module caméra, actif à "1". En mode "reset", le bus ZV-port produit par la caméra C38A est mis en haute impédance et est alors utilisé en partie pour lire la DRAM depuis le PC.
- D[7..0] : bus de données "luminance" bi-directionnel
- D[15..8] : bus de données "chrominance" bi-directionnel
- /RAS[3..0] : échantillonnage des adresses lignes (ici VL[9..0]) au ↓, dans les 4 bancs de mémoire du module SIMM (4x8bits=32 bits). En fait, /RAS0 et /RAS1 suffiraient car on utilise que D15 à D0.
- /CAS : échantillonnage des adresses colonnes (ici HP[9..0]) au ↓. Provoque l'écriture ou la lecture des données si /RASx est à "0".

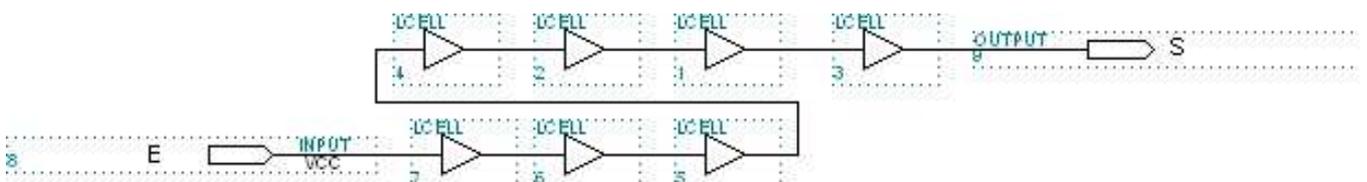
2.3 Signal vidéo N&B composite

La sortie CNA[7..0] et le signal CLK_CNA pilotent un convertisseur A/N pour produire sur le connecteur VQ_OUT de la maquette un signal vidéo N&B analogique "presque" conforme. En fait la synchro trame ne respecte pas le standard mais est généralement acceptée par les moniteurs.

On peut ainsi observer l'image à mémoriser en phase "écriture" et l'image mémorisée en phase "lecture".

3. Analyses des fonctions

3.1 Retard



Elle est simplement constituée de 7 cellules "LCELL" (utilisées ici comme fonctions "Retard") permettant de retarder le signal PCLK et ainsi obtenir un échantillonnage correct des composantes numériques Y[] et UV[] dans la fonction "Driver_ZV_port_dram". En effet, l'EPLD utilisée sur la maquette d'étude est la plus lente de sa catégorie : EPF10K20RC240-4; les temps de propagation de base sont de l'ordre de 15nS, à comparer avec la période de PCLK : 74nS (13,5MHz).

De plus, le signal PCLK produit par la caméra a des temps de commutation assez longs.

Le nombre de cellules a été déterminé expérimentalement en observant à l'oscilloscope mixte les signaux CLK_CNA (sortie de "Retard") et Y[7..0] de sorte que le ↑ encadre des infos stables.

3.2 Driver_ZV_port_dram

C'est la fonction principale. Elle a en charge la gestion de la RAM dynamique :

- écriture d'une image en couleur de 640x480 pixels
- lectures séquentielles de la luminosité et de la couleur des ces 307200 pixels

et la production du signal /CSYNC (synchro composite simplifiée).

```
subdesign DRIVER_ZV_PORT_DRAM
(
  RST      : input;  -- Reset du chip OV7620 : "0": caméra active, "1":lecture DRAM
  Y[7..0]  : input;  -- Composante luminance bus ZV Port
  UV[7..0] : input;  -- Composante chrominance bus ZV Port
  PCLK     : input;  -- Horloge "pixels" ZV Port (13,5MHz)
  HREF     : input;  -- Validation horloge "pixels"
  VSYNC    : input;  -- Synchro V. Passe à "1" pdt l'intervalle D du standard CCIR
              -- Utilisé pour initialiser HP et VL en lecture (RST="1")
  SEL_Y_UV : output; -- Pour sélectionner Y ou UV ds la DRAM en phase lecture
  YO[7..0] : output; -- Composante luminance : 13,5M échantillons/s
  VU[7..0] : output; -- Composantes U & V alternées codées C2 :
              -- 13,5M échantillons/s
  /RAS     : output; -- Pour piloter RAM dynamique
  /CAS     : output; -- Pour piloter RAM dynamique
  AM[9..0] : output; -- Adresse RAM multiplexées
  /CSYNC   : output; -- Synchronisation composite "simplifiée"
  TRAME    : output; -- Signal FODD reconstitué : "1"-> trame impaire
)
variable
YO[7..0], VU[7..0] : DFF; -- Composantes échantillonnées du port ZV
HP[9..0] : DFF; -- Compteur pixel H modulo 858 si caméra active
              -- 640 si lecture DRAM
VL[9..0] : DFF; -- Compteur ligne modulo 525 si caméra active
              -- "lignes actives" (480) si lecture DRAM
H_START[1..0] : DFF; -- Pour détecter mise à "1" de HREF ou RST
VSYNC_FF : DFF; -- Synchronisation VSYNC sur PCLK pour éviter les aléas
/CSYNC : DFF; -- Synchronisation composite "simplifiée"
TRAME : DFF; -- Trames paire("0")/impaire("1"). FODD reconstitué
START_I : DFF; -- Start Image : pour affecter VL[] à la mise à "1" de TRAME
/RAS_FF : DFF; -- Pour produire /RAS
SEL_Y_UV : DFF; -- Pour sélectionner Y ou UV ds la DRAM en phase lecture

begin
-- *****
-- Echantillonner les composantes et conversion C2 sur UV
-- *****
YO[].d=Y[]; -- Composante Y codée en BN
YO[].clk=PCLK and HREF; -- Un échantillon Y[] par période qd HREF="1"
VU[].clk=PCLK and HREF; -- Un échantillon UV[] par période qd HREF="1"
VU[6..0].d=UV[6..0]; VU7.d=!UV7; -- VU[] converti en C2

-- *****
-- Production des signaux TRAME et START_I
-- *****
VSYNC_FF.clk=PCLK; VSYNC_FF.d=VSYNC;-- Synchronisation VSYNC pour éviter aléas
TRAME.clk=VSYNC_FF.q;
TRAME.d=HP9.q; -- Au flanc montant de VSYNC, HP9 identifie la trame :
              -- "1" -> impaire ou "0" -> paire
START_I.clk=PCLK; -- Start Image
START_I.d=TRAME.q; -- Pour détecter mise à "1" de TRAME
              -- => VL[] <- 0 en mode caméra
              -- VL[] <- 19 en mode lecture DRAM

-- *****
-- Production H_START[]
-- *****
H_START[].clk=PCLK;
if !RST
  -- Caméra active
  then H_START0.d=HREF; -- Pour détecter mise à "1" de HREF
  -- Caméra inactive et lecture DRAM
  else H_START0.d=RST and HP[]<638;-- H_START0 passe à "1" sur flanc montant de PCLK
              -- qd RST passe à "1" (début lecture)
              -- et revient à "0" qd HP[]=638
              -- (avant dernier pixel H de chaque ligne)
end if;
H_START1.d=H_START0.q; -- H_START1 copie H_START0 avec 1 PCLK de retard
```

Application maquette d'étude EP10K20 – Caméra C38A, mémoire d'image, ZV port

```
-- *****
-- Compteurs HP[] et VL[]
-- *****
HP[].clk=PCLK; VL[].clk=PCLK;
-- RST="0" : caméra active
-- *****
-- - HL[] incrémenté modulo 858 à chaque flanc montant de PCLK
--   mis à 0 synchrone sur PCLK à la mise à "1" de HREF
-- - VL[] incrémenté modulo 525 au flanc montant de PCLK qd HL[]=857
--   mis à 0 asynchrone à la mise à "1" de TRAME
if !RST then if (HREF and !H_START0.q) or (HP[].q==857) -- Mise à "1" de HREF ou HP=857?
  then HP[].d=0; -- Oui -> Raz HP[]
  if VL[].q==524 then VL[].d=0; -- Si VL[]=524 -> VL[]=0
  else VL[].d=VL[].q+1;-- sinon incrémentation VL[]
  end if;
  else HP[].d=HP[].q+1; -- Non -> incrémentation HP[] à chaque pixel
  if TRAME.q and !START_I.q then VL[].d=0; -- Raz VL[] si début image
  else VL[].d=VL[].q;-- sinon maintien VL[]
  end if;
end if;
-- RST="1" : caméra inactive et lecture DRAM
-- *****
-- - HL[] incrémenté modulo 640 à chaque flanc montant de PCLK
-- - VL[] incrémenté modulo "lignes actives" au flanc montant de PCLK qd HL[]=639
--   (lignes actives : 19 à 258 (trame impaire) et 282 à 521 (trame paire))
-- affectation synchrone à 19 à la mise à "1" de RST
else if VSYNC -- Si RST="1" : lecture
  then HP[].d=0; VL[].d=519;-- Si VSYNC="1" de RST : Raz HP[] et VL[]<-19
  else if HP[].q==639 -- Dernier pixel d'une ligne ?
  then HP[].d=0; -- Oui -> HP[] passe à 0
  -- VL[] s'incrémte modulo "lignes actives"
  if VL[].q==19+239 then VL[].d=263+19; -- VL[] passe de 258 à 282
  else if VL[].q==263+19+239
  then VL[].d=19; -- Et de 521 à 19
  else VL[].d=VL[].q+1;
  end if;
  end if;
  else if !/RAS
  then HP[].d=HP[].q+1;--Incrémentation HP[]: pixel suivant d'une ligne
  else HP[].d=HP[].q;-- Blocage de HP[] qd /RAS="0" (cycle RAS en cours)
  end if;
  VL[].d=VL[].q; -- Blocage VL[] pdt balayage ligne
end if;
end if;

-- *****
-- Production SEL_Y_UV
-- *****
-- Utilisé uniquement en mode lecture pour sélectionner alternativement les composantes Y et UV
SEL_Y_UV.clk=PCLK;
if !RST then SEL_Y_UV.d=B"0"; -- Non utilisé en mode écriture (caméra active)
  else if HP[].q==639 and VL[].q==521 then SEL_Y_UV.d=!SEL_Y_UV.q;
  else SEL_Y_UV.d= SEL_Y_UV.q;
  end if;
end if;

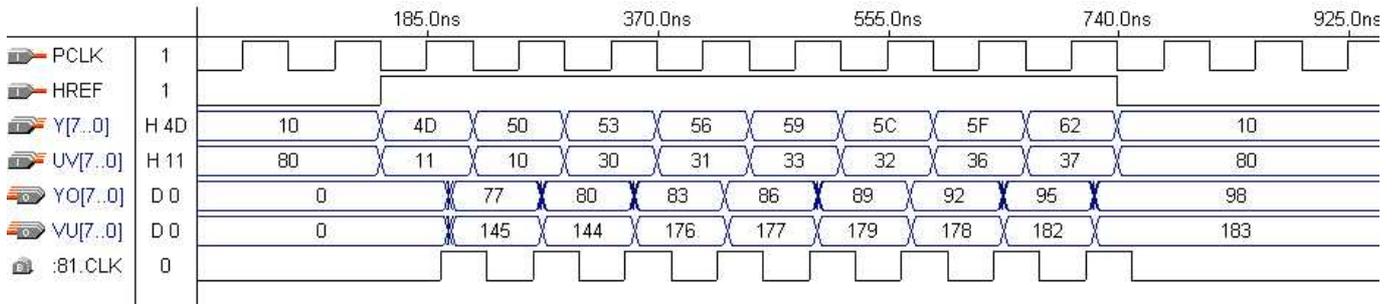
-- *****
-- Production /CAS et /RAS
-- *****
-- On utilise le mode EDO : un seul échantillonnage RAS pour 639 CAS
-- La DRAM utilisée doit comporter au moins 1024 colonnes
-- Voir les chronogrammes de simulation
-- Le flanc descendant de /RAS identifie VL[] sur AM[]
-- Le flanc descendant de /CAS identifie HP[] sur AM[]
/RAS_FF.clk=!PCLK; /RAS_FF.d=!H_START0;
/RAS=/RAS_FF and !(H_START0.q and H_START1.q);
/CAS=(H_START1.q and !PCLK);

-- *****
-- Multiplexage adresse DRAM
-- *****
-- Voir les chronogrammes de simulation
if H_START1.q then AM[]=HP[];
  else AM[]=VL[];
end if;
-- *****
-- Production /CSYNC
-- *****
/CSYNC.clk=PCLK;
/CSYNC.d=(VSYNC or (HP[]>=720+19-25 and HP[]<=720+19+52-25));
end;
```

3.2.1 Echantillonner les composantes luminance et chrominance

Il s'agit de mémoriser les états des signaux Y[7..0] et UV[7..0] au \uparrow de PCLK quand HREF="1". Ceci est facilement réalisé avec 2 registres de 8 bascules D : YO[7..0] et VU[7..0].

Le résultat de simulation ci-dessous illustre le bon fonctionnement de la structure. Le signal ":81.CLK" correspond au signal d'horloge (PCLK and HREF) des 16 bascules D.

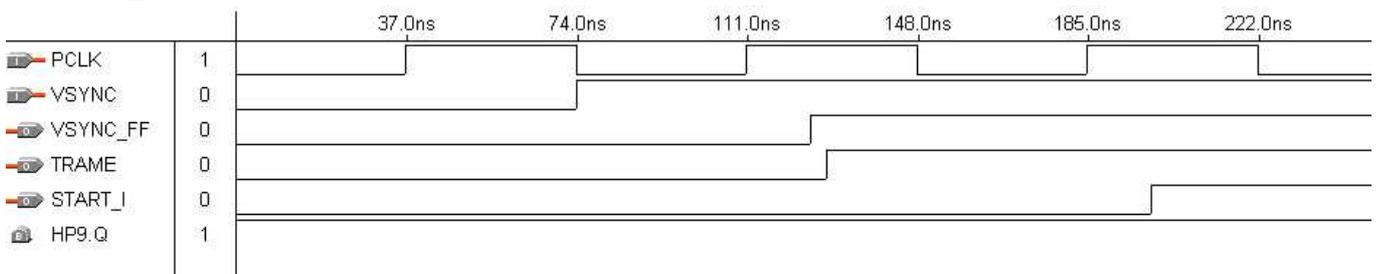


Le temps de propagation à partir des \uparrow de PCLK est de 20nS environ.

Note : la durée de HREF dans cette simulation n'est pas conforme au cas réel : HREF dure en fait 640 périodes PCLK !

3.2.2 Production des signaux TRAME et START_I

On commence par produire le signal VSYNC_FF qui est la copie "synchronisée" de VSYNC pour éliminer les flancs de montée parasites de VSYNC. Le faible retard (74nS max) entre VSYNC et VSYNC_FF est sans effet.



Comme on le verra plus loin, l'état du signal HP9 (bit 9 de l'adresse pixel H) au moment du \uparrow de VSYNC (donc aussi de VSYNC_FF) identifie la trame qui démarre : "1" \rightarrow impaire et "0" \rightarrow paire. Le signal TRAME est alors facile à produire avec une simple bascule D qui mémorise l'état de HP9 au \uparrow de VSYNC_FF. En fait, TRAME est la reconstitution du signal FODD produit par la caméra.

Le signal START_I (pour Start Image) est la copie de TRAME retardé d'une période PCLK. Il est utilisé en combinaison avec TRAME pour produire un signal logique qui passe à "1" pdt une période PCLK au début de chaque image : "TRAME and not START_I".

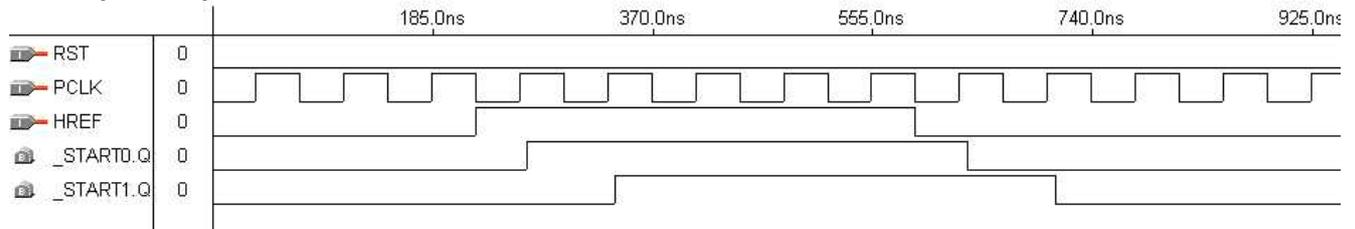
Ce signal est utilisé pour initialiser le compteur VL[9..0] (voir étude compteurs HP[] et VL[]).

3.2.3 Production du signal H_START[1..0]

Les signaux H_START[] permettent d'identifier le premier pixel "actif" de chaque ligne dans la mémoire d'écran quand elle est balayée en écriture ou en lecture.

Le mode de production de ces signaux dépend de l'état du signal RST :

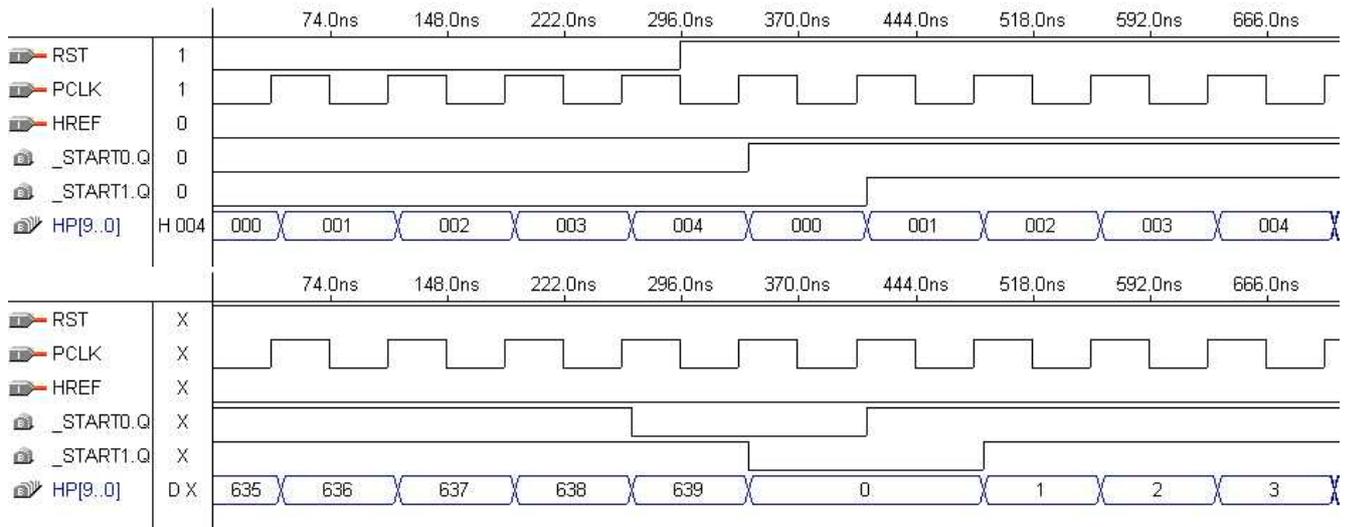
- "0" : caméra active, soit écriture dans la mémoire d'écran → H_START est produit à partir du signal HREF produit par la caméra.



La combinaison "HREF and not H_START0" identifie la mise à "1" de HREF.

Note : la durée de HREF ne correspond pas à un cas réel.

- "1" : caméra inactive, soit lecture de la mémoire d'écran → H_START est produit à partir du signal RST et de l'état du compteur HP[.].

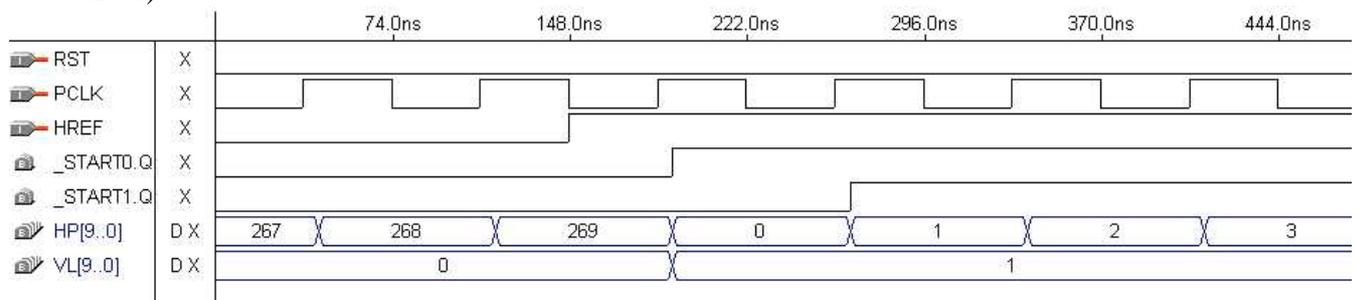


Les signaux H_START sont utilisés pour produire /RAS et /CAS. Les chronogrammes seront justifiés dans le paragraphe les concernant.

3.2.4 Compteurs HP[7..0] et VL[7..0]

Les états des compteurs HP et VL sont utilisés pour adresser la mémoire d'écran et produire /CSYNC :

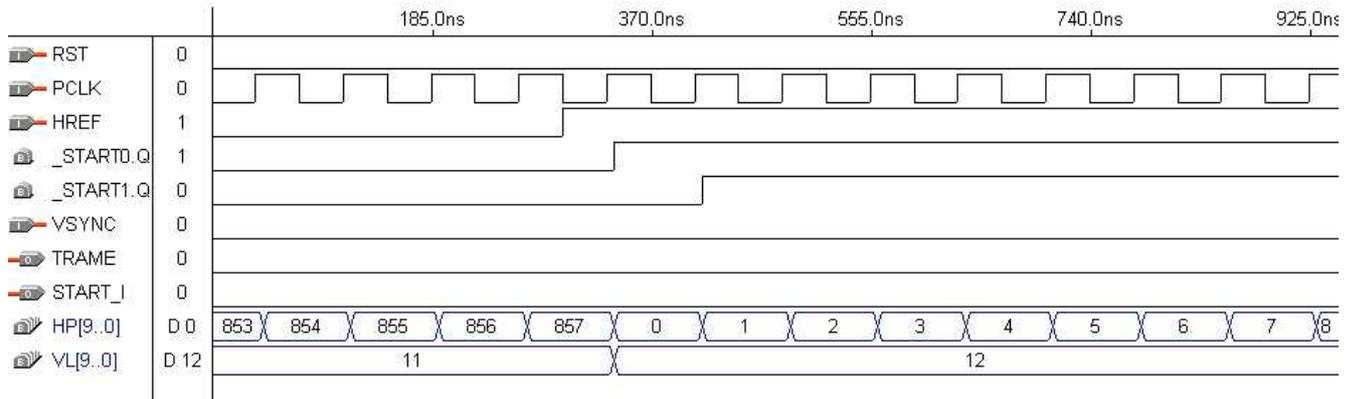
- **En mode écriture (RST="0")**
 - HP[9..0] : compteur incrémenté par PCLK modulo 858 (car 858 périodes PCLK = durée d'une ligne NTSC = 63,555µS), mis à 0 pour le premier pixel actif (qd HREF passe à "1") de chaque ligne.
Note : ainsi initialisé, l'état de HP correspond à la numérotation des pixels de la norme BT.601
 - VL[9..0] : compteur synchrone sur PCLK, incrémenté quand HP[]=857 modulo 525 (dans le format NTSC, une "image" comporte 525 lignes, mais la caméra ne produit que 480 lignes actives)



HP s'incrémente à chaque ↑ de PCLK et est bien mis à 0 de façon synchrone sur PCLK qd HREF passe à "1".

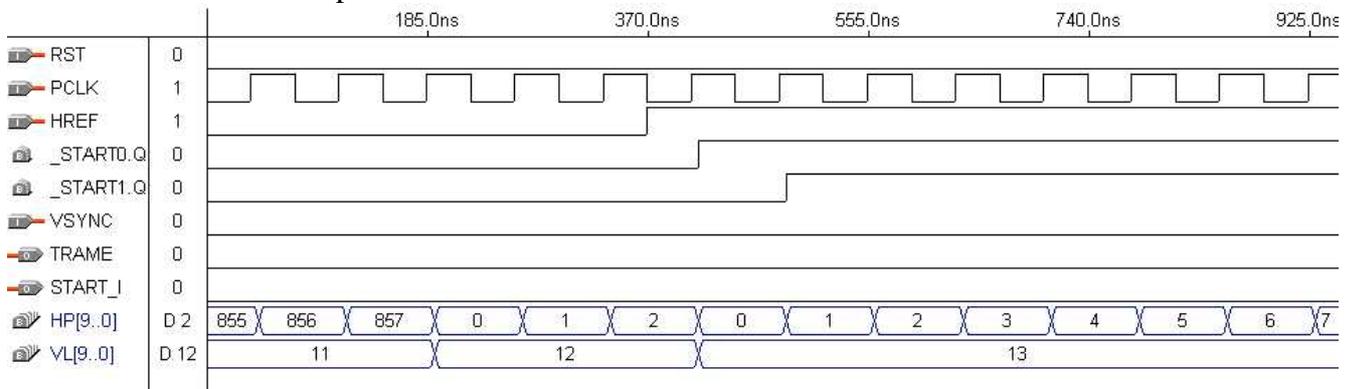
Application maquette d'étude EP10K20 – Caméra C38A, mémoire d'image, ZV port

VL s'incrémente quand HP est mis à "0", donc au changement de ligne. Pour les autres ↑ de PCLK, le compteur VL reste dans son état. Tout est conforme.



Dans le chronogramme ci-dessus, le signal HREF est conforme au cas réel (durée = 858 périodes PCLK). Le compteur HL est donc mis à 0 pour 2 causes concomitantes : HREF est passé à "1" et l'état de HL a atteint son maximum (857 car modulo 858).

L'incrémentation du compteur VL est conforme.



Dans le chronogramme ci-dessus, HREF et l'état de HP ne sont pas "verrouillés" en phase. Les compteurs HP et VL s'incrémentent quand HP vaut 857, mais sont également affectés quand HREF passe à "1", qq instants plus tard. Mais la période de HREF est exactement de 858 périodes PCLK (une ligne), de sorte que l'état de HP sera 857 à la prochaine mise à "1" de HREF (chronogramme précédent) et les signaux sont "verrouillés".

En mode lecture : RST = "1"

L'état du signal RST est contrôlé par le PC et son activation provoque le "reset" de la caméra. Elle ne produit alors plus aucun signal et toutes les broches sont mises en "haute impédance".

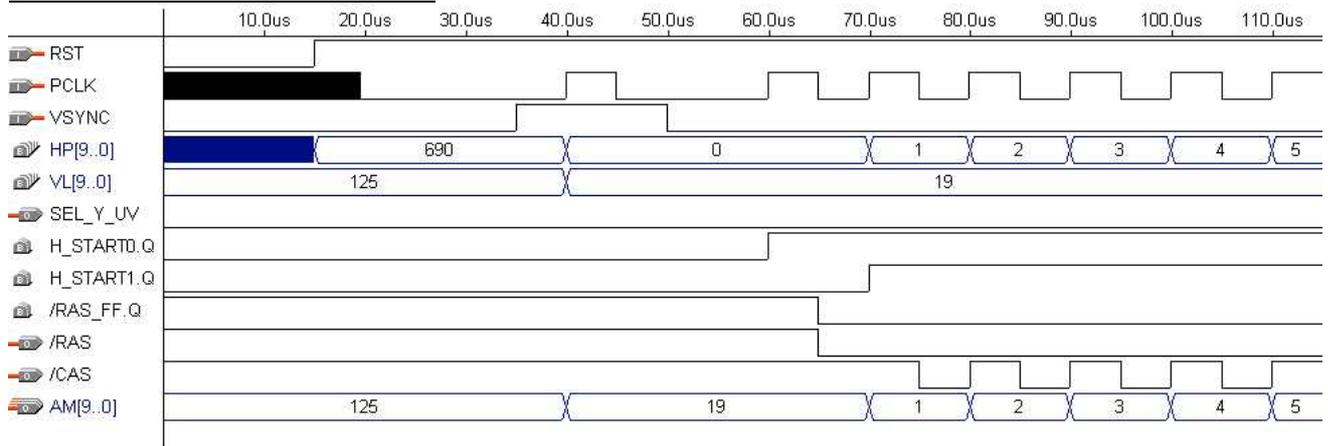
Le signal d'horloge PCLK est maintenant produit par le PC et son rythme baisse considérablement car on utilise le port imprimante. De plus, on ne peut pas parler de fréquence car le système d'exploitation Windows (ou Linux) provoque des "pauses".

Le signal VSYNC est produit par le PC et passe à "1" pdt la première impulsion PCLK pour initialiser les structures liées à la lecture (notamment les compteurs HP et VL).

En mode lecture, le signal PCLK incrémente toujours les compteurs ligne (HP) et colonnes (VL) pour balayer les pixels de l'image, mais pour gagner du temps on "saute" les pixels inactifs.

L'évolution des compteurs HP et VL permet d'adresser séquentiellement les 640 pixels de chacune des 480 lignes "utiles" sans temps mort.

Simulation du début de la lecture



Quand RST est à "0", PCLK est produit par la caméra et est de fréquence 13,5MHz. L'échelle de ce chronogramme ne permet pas de distinguer les périodes.

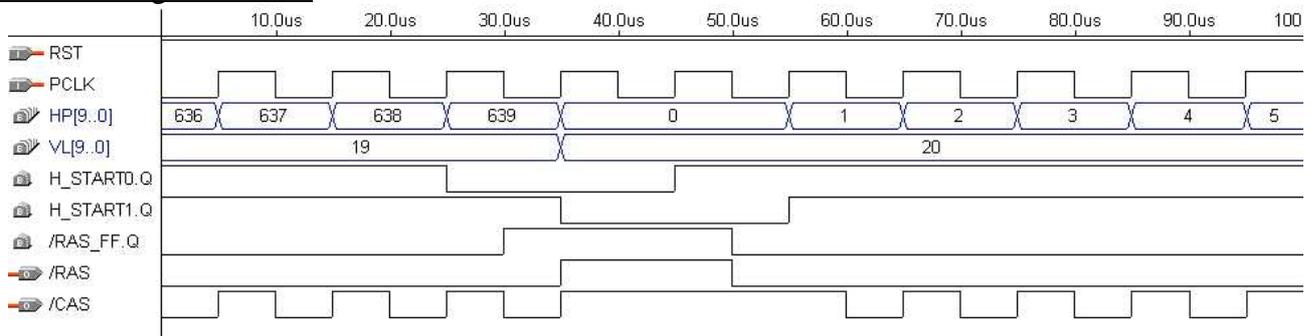
RST est mis à "1" par le PC, ce qui rend la caméra inactive et le signal PCLK est alors contrôlé par le logiciel exécuté par le PC. Les états de HP et VL à ce moment sont quelconques.

Le rythme de PCLK baisse alors considérablement et pour cette simulation, on a choisit une période de 10µs, proche du cas réel.

Le signal VSYNC produit par le PC valide les initialisations nécessaires. Au ↑ de PCLK qd VSYNC=1 :

- HP[9..0] ← 0 : on adresse le premier pixel d'une ligne
- VL[9..0] ← 19 : la première ligne active est la 20^{ième} depuis le début de l'image
- SEL_Y_UV ← 0 : on commence par la lecture de la composante Y
- H_START[1..0] ← 0 : voir étude production /RAS et /CAS

Passer à la ligne suivante

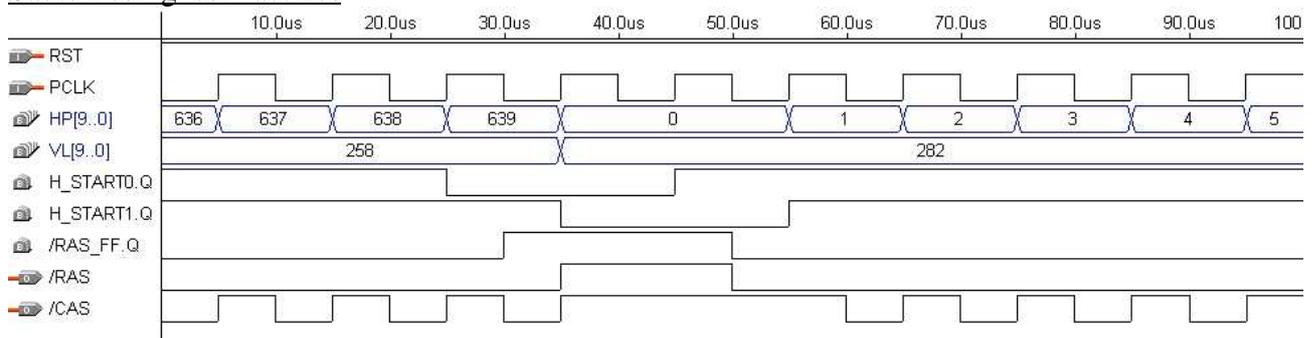


Quand HP[]=639, on constate bien que HP[] passe à 0 et que VL[] est incrémenté au prochain flanc montant de PCLK.

Le compteur VL reste dans son état pour tous les autres ↑ de PCLK.

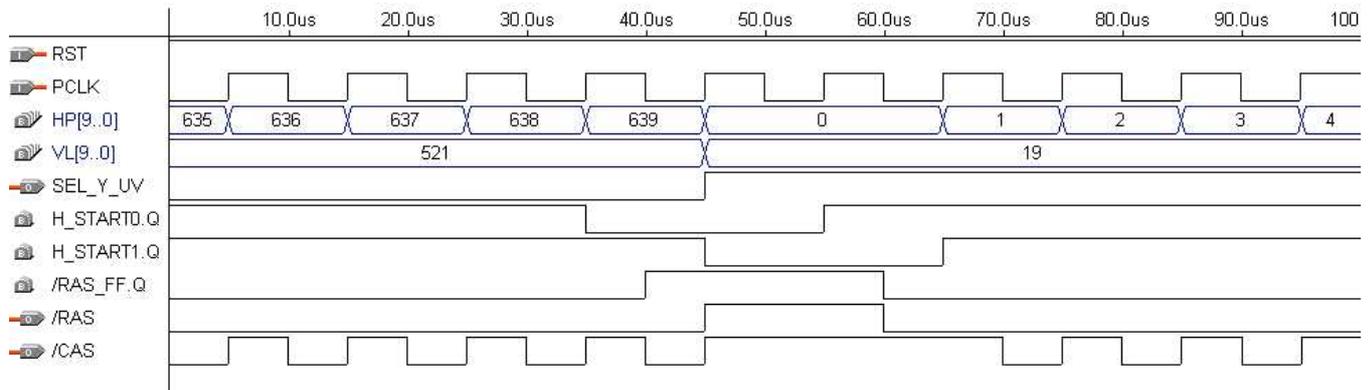
Le signal /RAS est à "1" au ↑ suivant de PCLK (voir plus loin) : HP[] garde son état (0) conformément à la description.

Sauter les lignes inactives



On passe bien du dernier pixel de la ligne 258 (dernière ligne active de la trame impaire) au premier pixel de la ligne 282 (première ligne active de la trame paire).

La chronologie des autres signaux est identique.



On passe bien du dernier pixel de la ligne 521 (dernière ligne active) au premier pixel de la ligne 19 (première ligne active de la trame impaire).

3.2.5 Production signal SEL_Y_UV

Ce signal est utilisé uniquement en mode lecture pour sélectionner alternativement les composantes Y et UV à la fin du balayage des 480 lignes utiles.

Il doit être initialement à "0" (sélection composante Y) et doit changer d'état ("1" : sélection composante UV) quand VL[] passe de 521 à 0. Cela est vérifié dans le chronogramme ci-dessus.

3.2.6 Production /RAS , /CAS et AM[9..0]

Ces signaux pilotent l'adressage de la RAM dynamique. Celle-ci est organisée en au moins 1024 lignes de 1024 colonnes et chaque cellule comporte au moins 16 bits (en fait les modules SIMM sont organisés en mots de 32 bits, ce qui est largement suffisant).

Pour limiter le nombre de broches, les adresses "lignes" et "colonnes" sont multiplexées sur un bus d'adresse commun nommé ici AM[9..0].

Les "chips" de DRAM comportent des registres pour mémoriser les adresses lignes et colonnes. Les signaux /RAS et /CAS sont utilisés pour échantillonner la bonne valeur dans le bon registre :

- /RAS = Row Adress Strobe : actif au ↓. L'adresse "ligne" (ici VL[9..0]) doit être appliquée aux broches A[9..0] du chip DRAM et doit rester stable de part et d'autre du ↓.
- /CAS = Colonne Adress Strobe : actif au ↓. L'adresse "colonne" (ici HP[9..0]) doit être appliquée aux broches A[9..0] du chip DRAM et doit rester stable de part et d'autre du ↓.

Les 2 signaux doivent être activés pour adresser complètement une cellule mémoire ce qui rend les cycles d'écriture et de lecture relativement longs. C'est une forte contrainte dans notre cas, car la vitesse d'écriture dans la DRAM est de 13,5Méchantillons par seconde.

Les cycles d'écriture et de lecture peuvent être réduit si les chips DRAM sont compatibles EDO, ce qui est pratiquement toujours le cas aujourd'hui.

En effet, si on accède **séquentiellement** aux 1024 cellules d'une **même ligne** (ce qui est le cas ici car cela correspond au balayage de la caméra) : **l'adresse colonne ne change pas**. Il devient alors inutile de répéter l'activation de /RAS pour ces cellules. Mais attention : toutes les DRAM ne permettent pas ce mode d'adressage.

Les chronogrammes de simulation présentés ci-dessous ont été réalisés en **mode "écriture"** (RST="0") qui est la phase la plus critique à cause de la vitesse élevée. En mode "lecture", les chronogrammes sont analogues (en fait plus lents) car les signaux /RAS et /CAS sont produits à partir de H_START[1..0] et PCLK quel que soit le mode. On peut se reporter aux derniers chronogrammes du §3.2.5 dans lesquels ont été inclus les signaux /RAS et /CAS.

La production de /RAS nécessite un signal intermédiaire nommé /RAS_FF. C'est la sortie d'une bascule D ayant comme signal d'horloge PCLK et comme donnée H_START0.

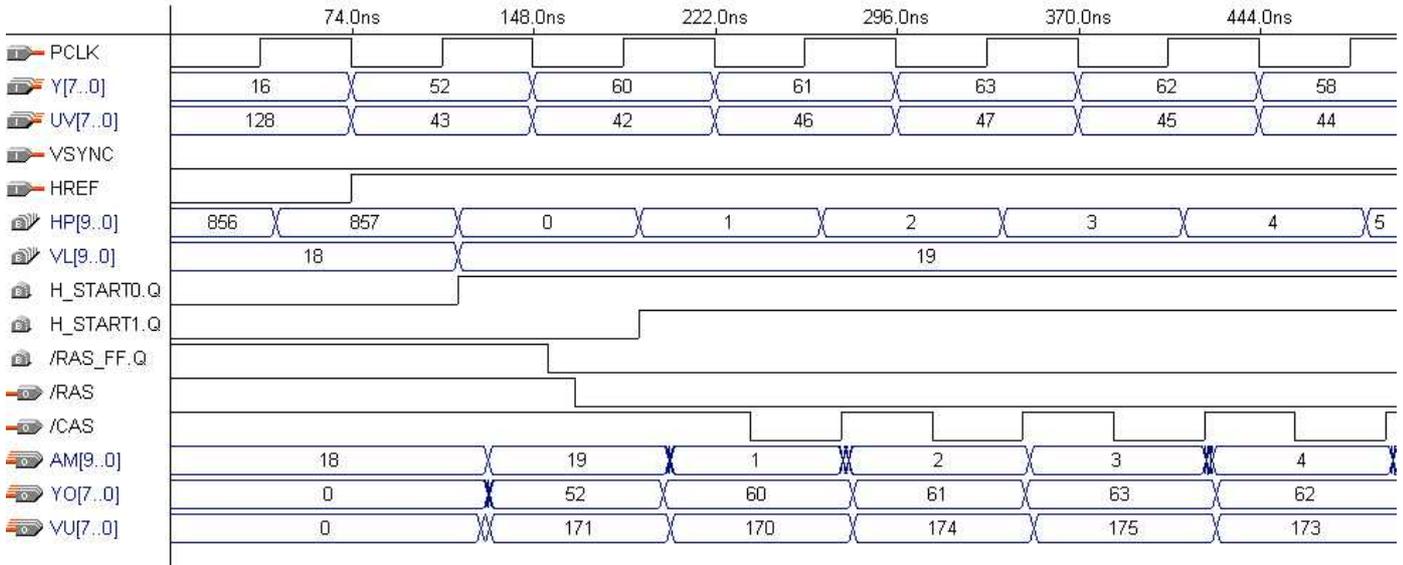
On constate dans le chronogramme ci-dessous, l'évolution conforme du signal /RAS_FF. En tenant compte, bien entendu, des temps de propagation non négligeables de l'EPLD.

Les équations booléennes des signaux /RAS et /CAS sont :

$$/RAS = /RAS_FF \text{ and not } (\text{not } H_START0 \text{ and } H_START1)$$

$$/CAS = \text{not } (H_START1 \text{ and not } PCLK)$$

Application maquette d'étude EP10K20 – Caméra C38A, mémoire d'image, ZV port



Dans la phase représentée ci-dessus, /RAS est simplement la copie de /RAS_FF car "not (not H_START0 and H_START1)" est toujours à "1". Le faible retard est dû aux temps de propagation. Le signal /CAS est conforme à son équation booléenne avec un léger retard dû à la même cause.

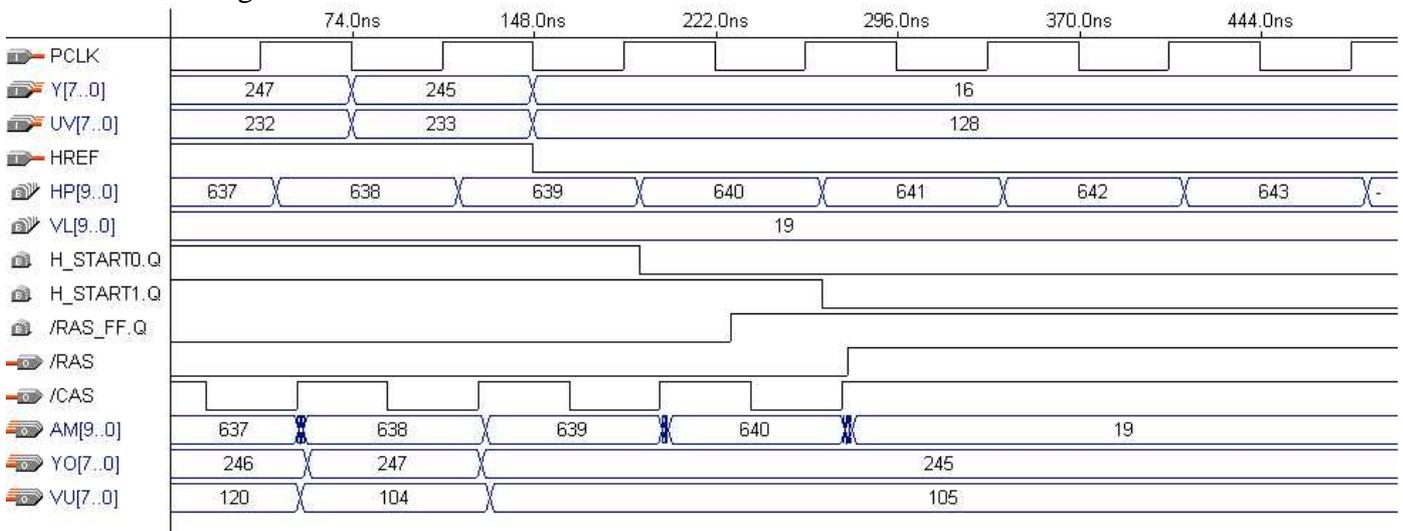
L'état du bus d'adresse multiplexé AM[9..0] est déterminé par le signal binaire H_START1 :

- H_START1 = "0" : AM[9..0] = VL[9..0]
- H_START1 = "1" : AM[9..0] = HP[9..0]

On constate que :

- AM[9..0]=VL[9..0] de part et d'autre du ↓ de /RAS : l'échantillonnage dans la DRAM des adresses "lignes" devrait se faire sans aléas, si la DRAM est assez rapide.
- AM[9..0]=HP[9..0] de part et d'autre du ↓ de /CAS : l'échantillonnage dans la DRAM des adresses "colonnes" devrait se faire sans aléas, si la DRAM est assez rapide.
- Les données à écrire dans la DRAM (YO[7..0] et VU[7..0]) sont stables avant le ↓ de /CAS et pendant l'état "0" de signal. Le cycle d'écriture est respecté (en mode EDO avec /RAS=0)
- Le premier pixel de chaque ligne n'est pas mémorisé (adresse colonne = 0). On perd un pixel sur 640 : ce n'est pas gênant.

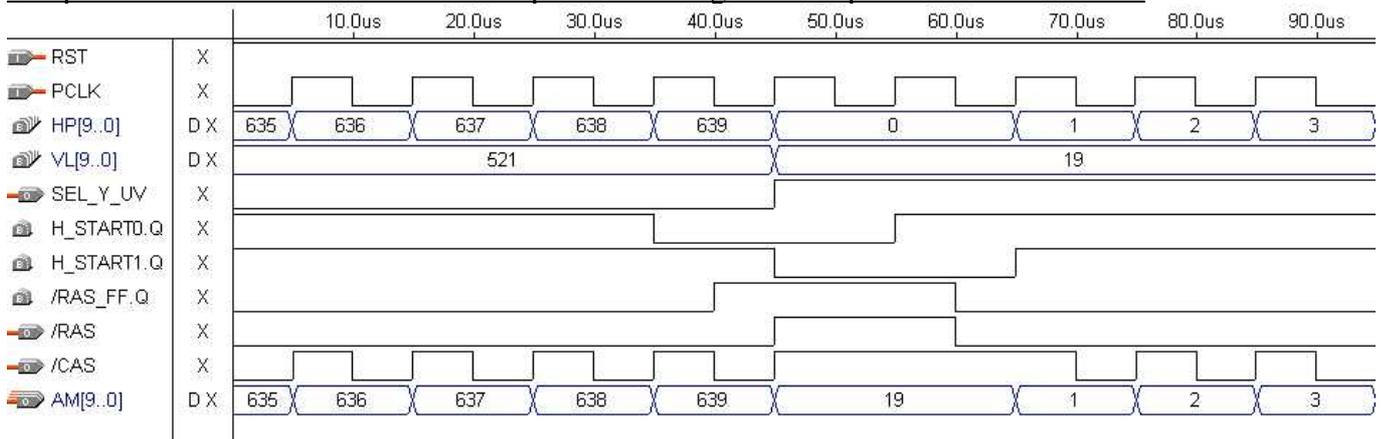
Le signal /RAS doit revenir à l'état "1" simultanément à /CAS pour être conforme avec les cycles EDO de la DRAM. Cette opération est réalisée à la disparition de HREF (fin partie active d'une ligne) comme l'illustre le chronogramme ci-dessous.



L'équation booléenne de /RAS permet son retour à l'état "1" simultanément à /CAS.

On mémorise un pixel "hors image" (adresse colonnes = 640). Il ne sera pas lu, ce n'est donc pas gênant.

Cas particulier de la lecture des derniers pixels d'une ligne et des premiers de la suivante



Cette simulation a été réalisée avec RST="1" et une période PCLK de 10µS.

On constate que :

- Les cycles de lecture sont conformes. La donnée est fournie par la DRAM juste après le ↓ de /CAS. C'est à ce moment qu'elle est lue par le PC via le bus Y[7..0].
- /RAS revient à "1" pour le dernier pixel utile d'une ligne (adresse colonnes = 639)
- Deux périodes PCLK sont nécessaires pour lire le prochain pixel de la ligne suivante. A noter que le pixel d'adresse colonne 0 n'est pas lu. Ce n'est pas gênant car il n'a pas été mémorisé !

3.2.7 Production du signal /CSYNC (synchro composite)

Ce signal permet, via la fonction "CNA_out", de produire un signal vidéo composite N&B presque conforme, en tous cas suffisamment pour assurer un bon fonctionnement du moniteur de contrôle. Il n'est donc utilisé qu'en mode "écriture", quand la caméra est active.

Il est normalement à "1" et passe à "0" :

- Pendant 4µS environ à chaque ligne pour identifier la synchro ligne
- Quand VSYNC="1" pour identifier la synchro trame. C'est un moyen très simpliste !

L'état du compteur HP[9..0] est utilisé pour déterminer l'intervalle de temps pendant lequel /CSYNC doit être à "0". On respecte les recommandations de la norme BT.601 : /CSYNC est à "0" quand HP[] est compris entre $(720+19-25)=714$ et $(720+19+52-25)=766$, soit :

- Pendant une durée de 52 périodes PCLK = 3,85µS
- Passe à "0" $(858-714)=144$ périodes PCLK = 10,66µS après le dernier pixel
- Revient à "1" $(858-766)=6,8µS$ avant le 1^o pixel

3.3 TRI_8 "3" et "4"

Ces buffers 3 états "tamponnent" les sorties YO[7..0] et VU[7..0] de la fonction Driver_ZV_port_Dram avant de les appliquer à la DRAM (D[15..0]). Le seul but est le contrôle de leur état "haute impédance" en cohérence avec le mode de fonctionnement de la DRAM : lecture ou écriture.

Ce mode est déterminé par le signal /WR. Le même signal est donc utilisé pour piloter ces buffers pour assurer un bon transfert des données et éviter les conflits :

- /WR = "0" : écriture DRAM ⇒ les buffers sont validés
- /WR = "1" : lecture DRAM ⇒ les buffers en haute impédance

Dans cette première version de l'application, le signal /WR est simplement activé (mis à "0") quand on appuie sur une des touches 1, 2 ou 3 du clavier décimal. L'inconvénient, c'est que l'on ne maîtrise pas les instants de début et de fin de la phase d'écriture, ils peuvent se produire au milieu d'une image et perturber alors la mémorisation de qq pixels. Mais dans la majorité des cas, il n'y a pas de problèmes; le moniteur N&B permet un contrôle rapide.

3.4 Mux2 et TRI_8 "14"

En phase "écriture" (RST="0"), la caméra fournit les informations luminance et chrominance sur les bus Y[7..0] et UV[7..0] pour être mémorisées dans la DRAM. Le buffer TRI_8 "14" doit donc être en haute impédance pour éviter les conflits.

En phase "lecture" (RST="1"), la lecture de la DRAM est pilotée par le PC via son port parallèle. Les 8 bits de données de ce port sont reliés au bus Y[7..0].

Or, la luminosité et la couleur de chaque pixel sont codés sur 16 bits et mémorisés sous ce format dans la DRAM. La lecture d'une image complète suit la séquence suivante :

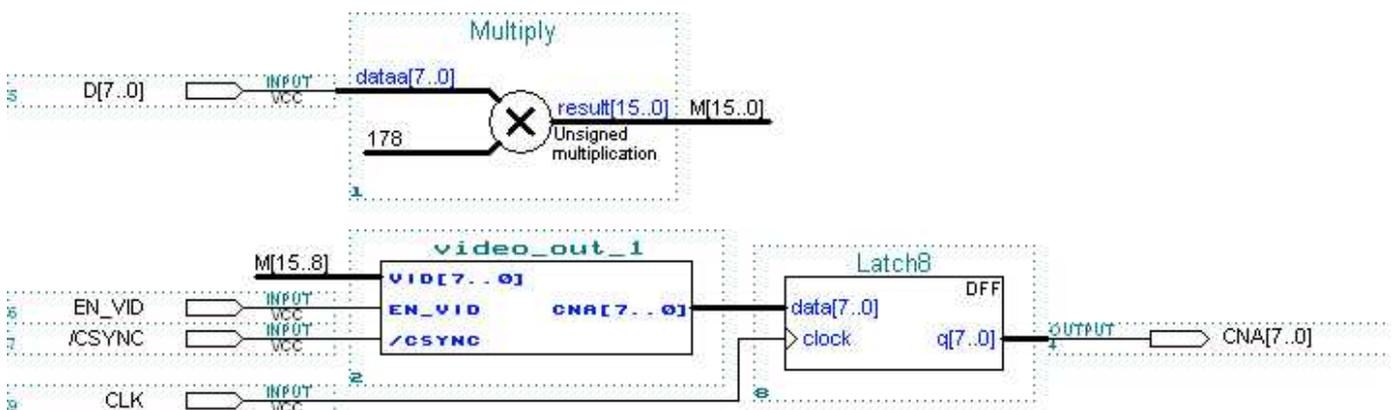
- RST="1"
- SEL_Y_UV="0" : lecture de la luminosité des 640x480 pixels de l'image mémorisés dans la DRAM (D[7..0]).
- SEL_Y_UV passe à "1" : lecture de la chrominance des 640x480 pixels de l'image mémorisés dans la DRAM (D[15..8]).

Le multiplexeur Mux2 réalise la sélection de D[7..0] ou D[15..8] en fonction de SEL_Y_UV et envoie le résultat sur le port Y[7..0] via le buffer TRI_8.

L'état du buffer MUX_8 "14" est simplement contrôlé par le signal RST.

3.5 CNA_out

La fonction combine l'information D[] codée en BN sur 8 bits (étendue : 16 à 252) et les signaux binaires EN_VID et /CSYNC pour produire un signal vidéo N&B composite presque conforme.



La structure de la sortie VQ_OUT de la maquette produit $V_{Q_OUT} = 0V$ qd $CNA[] = 0$ et $V_{Q_OUT} = 1V$ quand $CNA[] = 255$. Elle est suffisamment rapide pour suivre le rythme des échantillons ($13,5 \cdot 10^6/S$). Les valeurs affectées à CNA[] permettent de produire un signal vidéo analogique conforme : hauteur de la composante "synchro" = 0,3V ($CNA[] = 77$) et hauteur de la composante utile = 0,7V ($255 - 77 = 178$).

3.5.1 Multiply

L'équivalent décimal des 8 bits MSB du résultat (M[15..8]) varie de 0 à 177 quand celui de D[] varie de 0 à 255. Cette caractéristique est nécessaire à la fonction "Video_out_1".

A noter que l'implémentation de fonctions de multiplication semble simple avec Maxplus. Mais les structures associées sont très complexes car il s'agit de structures combinatoires. On peut ainsi facilement dépasser les ressources disponibles dans l'EPLD.

Mais elles ont la propriété d'être très rapides, ce qui est exploité ici : 13,5 millions de multiplications par seconde !

3.5.2 Video_out_1

```
subdesign VIDEO_OUT_1
(
  VID[7..0] : input; -- Luminance codée sur 8 bits : de 77 à 255
  EN_VID    : input; -- Fenêtres de validation
  /CSYNC    : input; -- Synchro composite
  CNA[7..0] : output;
)
begin
  if EN_VID then CNA[]=VID[]+77;
    else if /CSYNC then CNA[]=77;
      else CNA[]=0;
    end if;
  end if;
end;
```

3.5.3 Latch8

Ce registre permet de maîtriser l'instant d'échantillonnage en phase d'écriture où le rythme des échantillons est le plus élevé. De plus, le temps de calcul (de propagation en fait) de la fonction "Multiply" dépend des nombres traités.

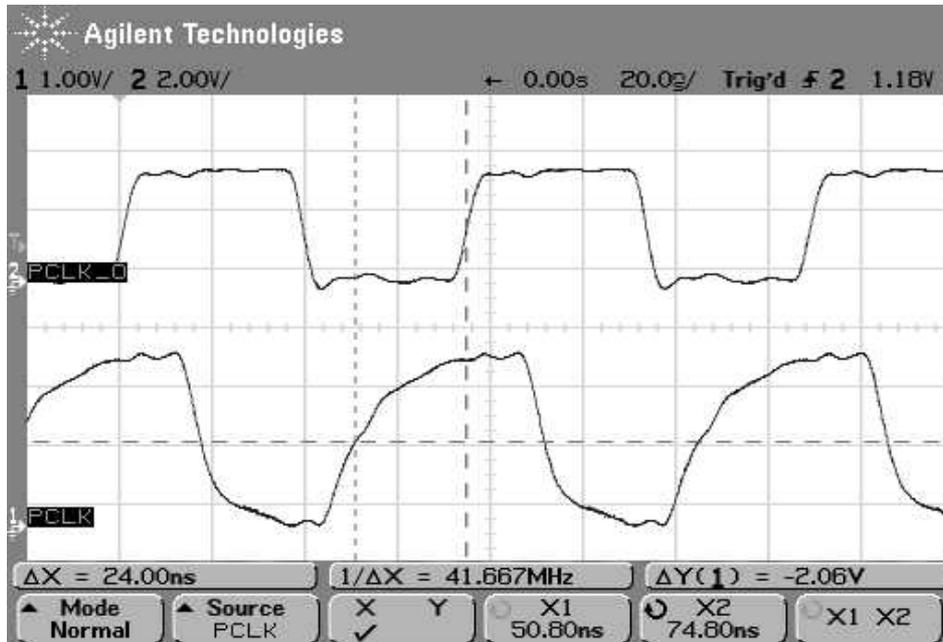
RELEVÉS EXPÉRIMENTAUX

Conditions :

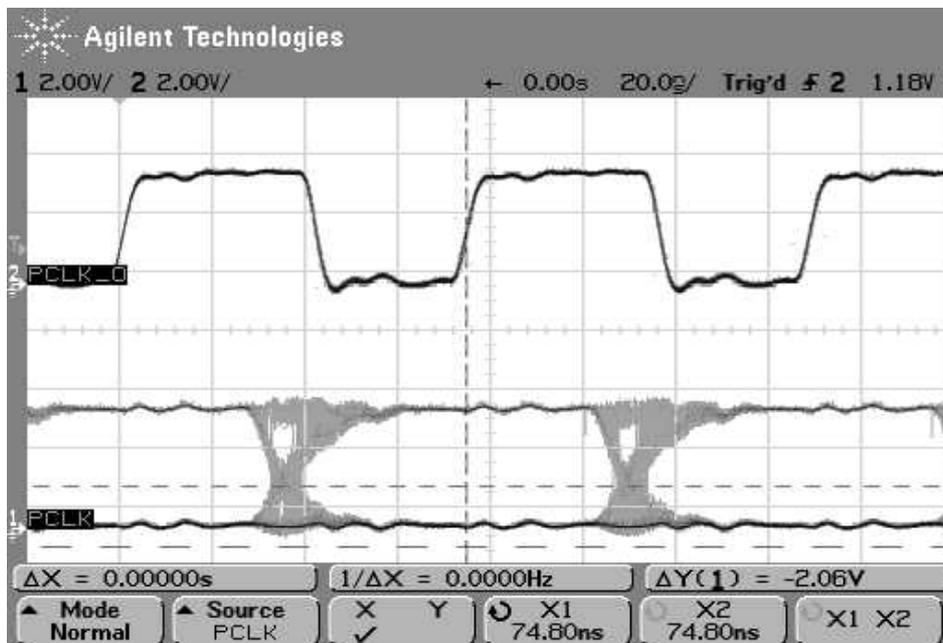
- Carte d'étude n° 1
- Caméra en mode NTSC 16 bits

1. Caméra active (RST="0")

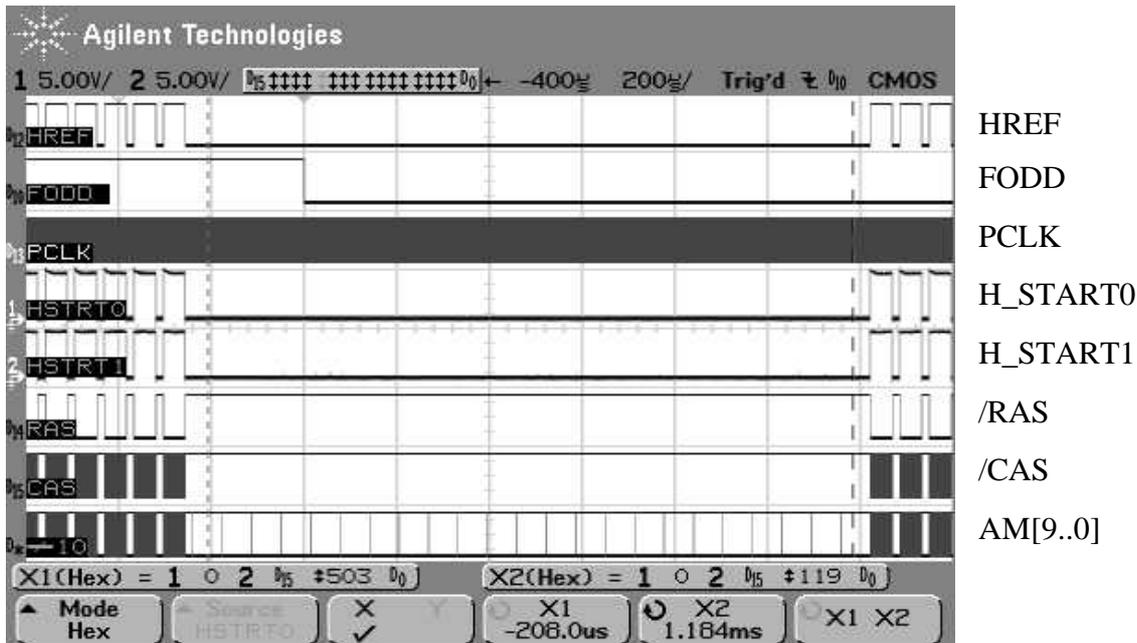
1.1 Mise en forme horloge PCLK



La fonction "Make_clock" retarde le signal d'horloge de 50nS environ pour un obtenir un diagramme de l'œil très satisfaisant. Le relevé ci-dessous le montre clairement.



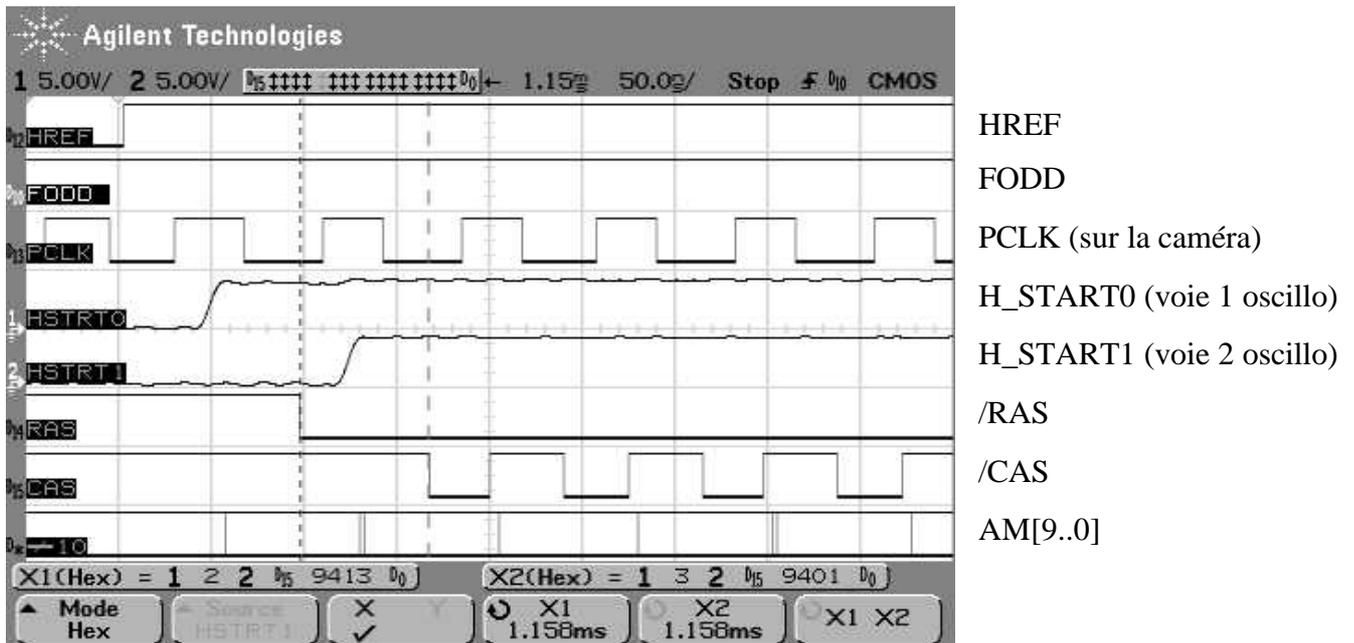
Même relevé dilaté autour du ↓ de FODD : fin de la trame impaire et début de la paire :



Le compteur VL est à l'état 103_H (259) au curseur X1. Cela signifie que la dernière ligne active correspond bien à l'état 258 de VL.

Le compteur VL est à l'état 119_H (281) au curseur X2. Cela signifie que la première ligne active correspond bien à l'état 282 de VL.

Premiers pixels de la première ligne utile :

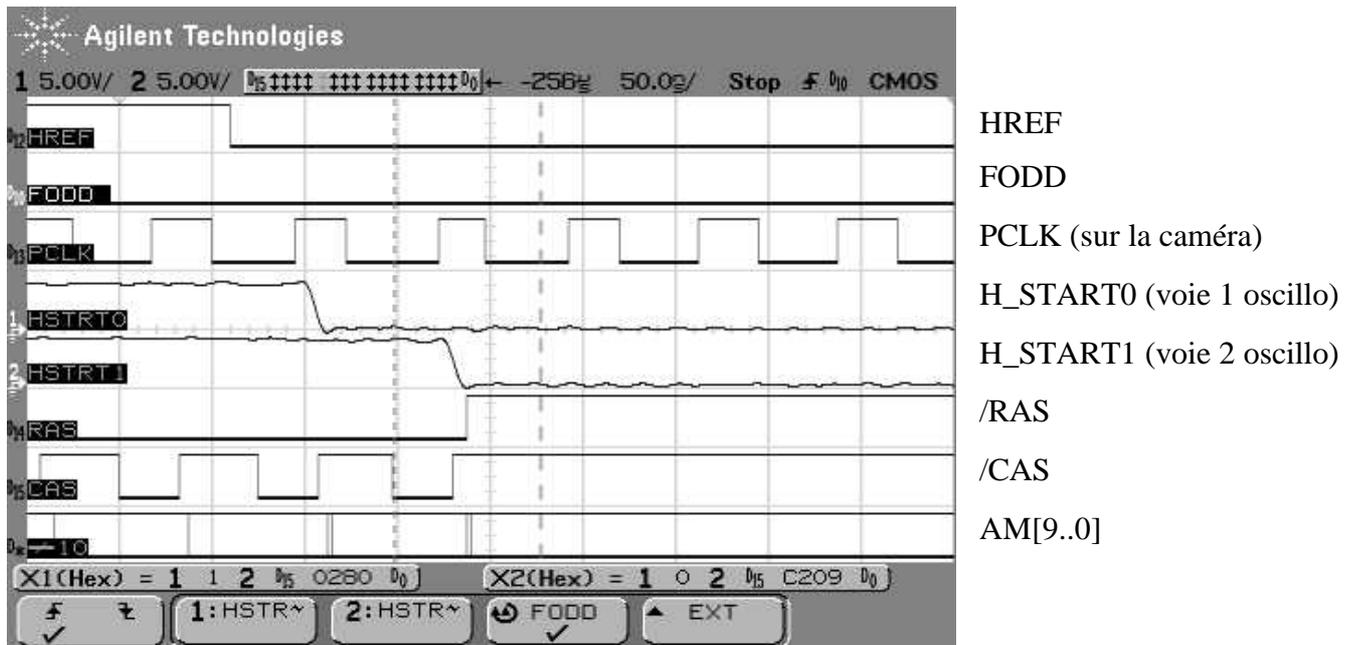


Rappel : AM[9..0]=HP[9..0] si H_START1="1" et AM[9..0]=VL[9..0] si H_START1="0"

On constate bien que :

- H_START0 copie HREF aux ↑ de PCLK,
- H_START1 copie H_START0 avec un retard de 1 période PCLK
- Les signaux /RAS et /CAS sont conformes au chronogramme de simulation du §3.2.7
- Au ↓ de /RAS : AM[9..0]=VL[9..0]=013_H=19 : CQFD. De plus le ↓ est bien centré et l'échantillonnage dans la DRAM est optimisé.
- Au ↓ de /CAS : AM[9..0]=HP[9..0]=001_H=1 : CQFD. Le ↓ est également bien centré.
- AM[9..0]=HP[9..0] s'incrémente alors régulièrement et chaque valeur est échantillonnée dans la DRAM aux ↓ de /CAS.

Derniers pixels de la dernière ligne utile de la trame paire :



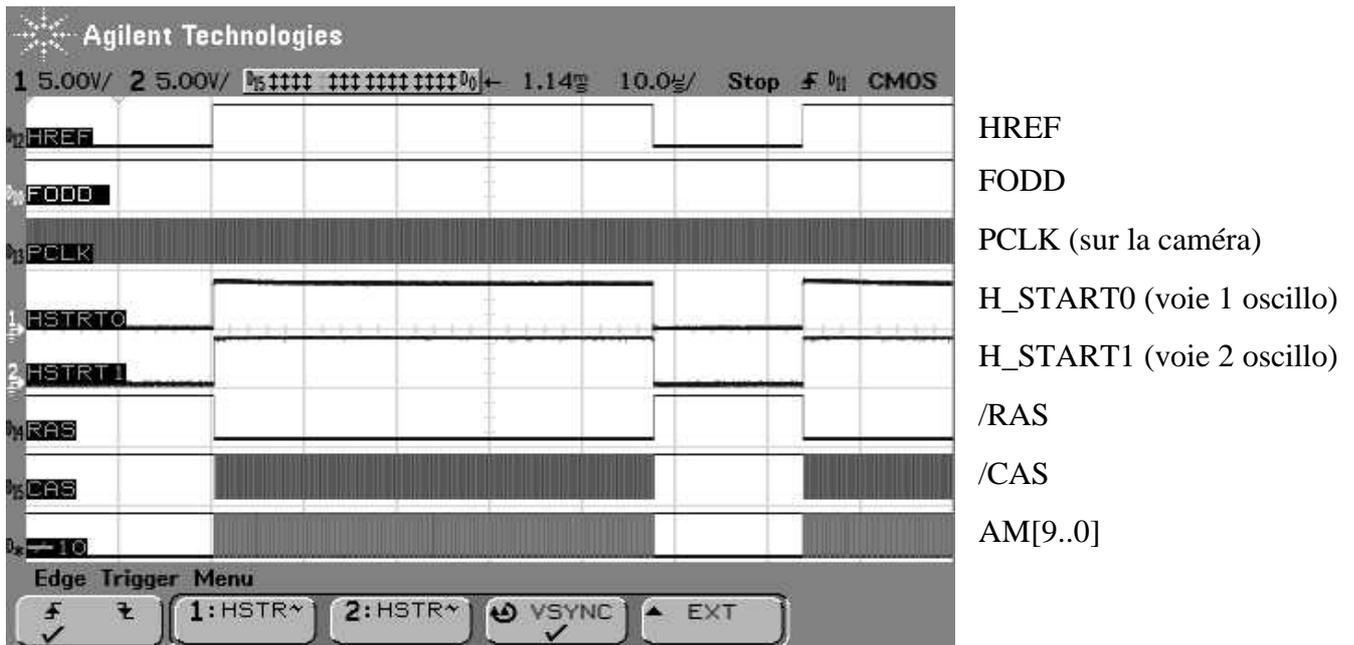
On constate bien que :

- Au ↓ de /CAS : AM[9..0]=HP[9..0]=280_H=640 : CQFD (voir §3.2.7).
- Juste après, H_START1 passe à "1" et AM[9..0]=HP[9..0]=209_H=521 : dernière ligne utile trame paire.

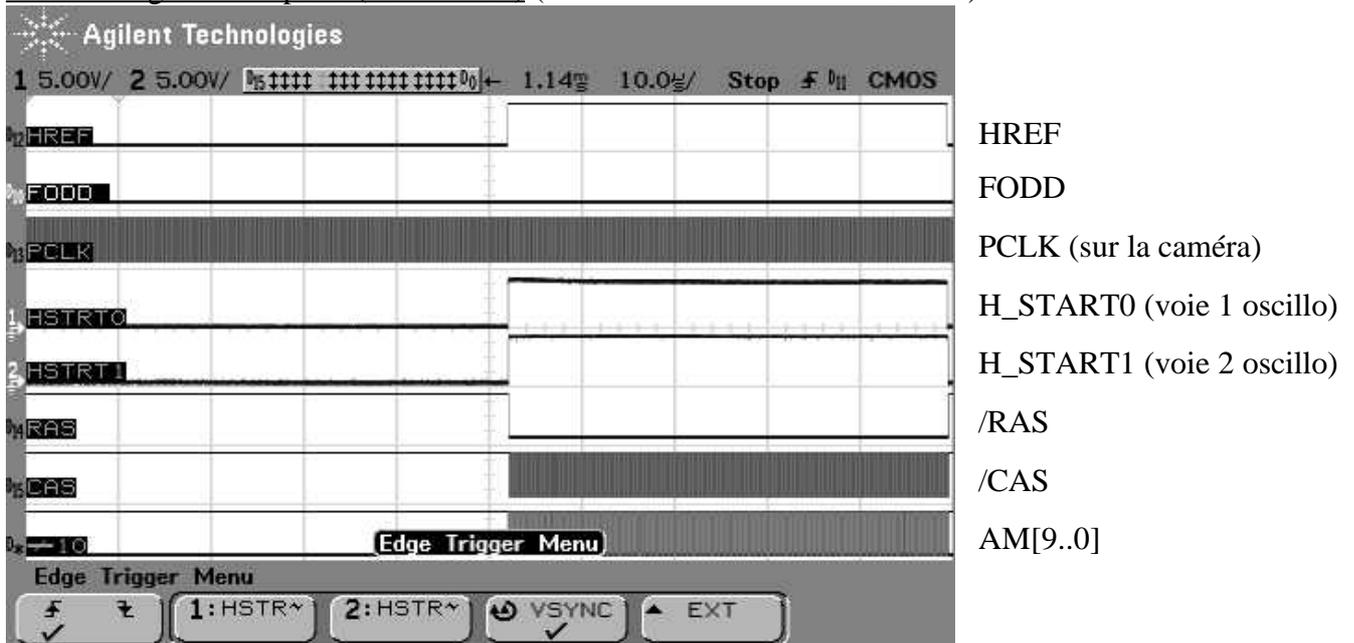
1.3 Production TRAME

Pour identifier les trames impaire et paire, on exploite la phase entre VSYNC et HREF:

Première ligne trame impaire (FODD="1") (déclenchement sur ↑ de VSYNC) :



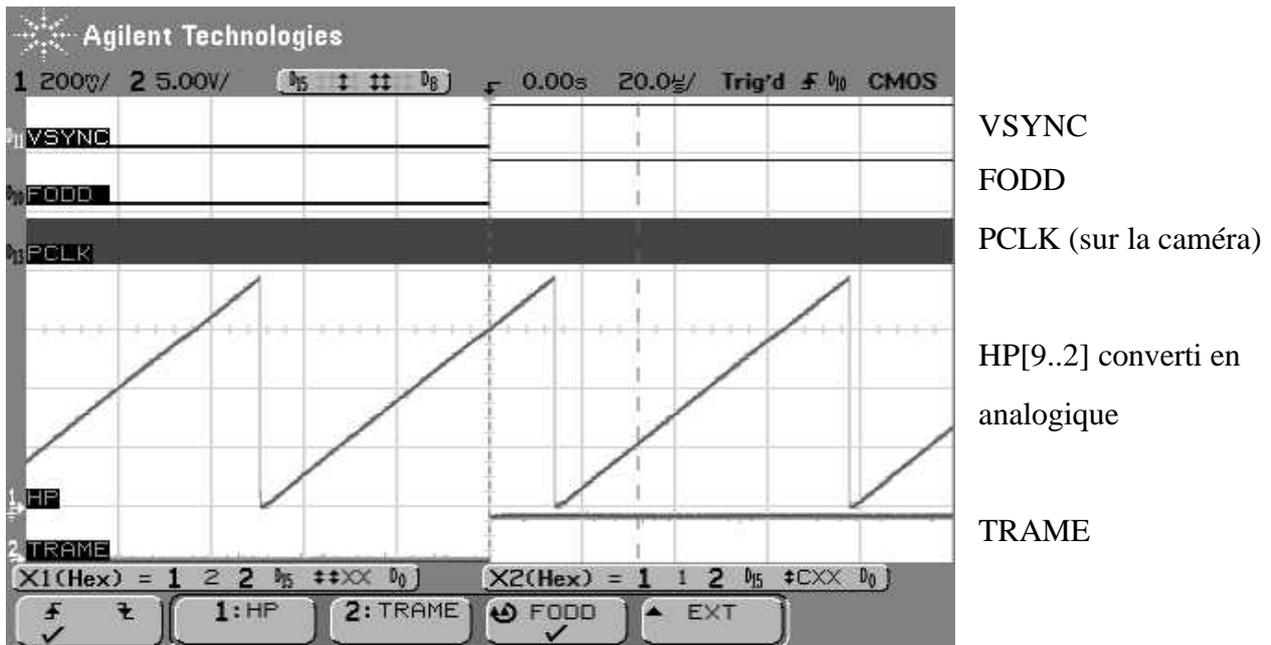
Première ligne trame paire (FODD="0") (déclenchement sur ↑ de VSYNC) :



On constate un retard entre les ↑ de VSYNC (déclenchement) et HREF plus long de 3,2 divisions (soit 32μS) dans le 2ième chronogramme. Ce décalage correspond exactement à durée d'une demi-ligne et provoque l'entrelacement des lignes impaires et paires sur l'écran.

L'état du compteur HP[9..0] représente à tout instant la position H sur l'écran : 0 à 857. On va donc l'utiliser pour identifier les trames impaire et paire et produire le signal TRAME.

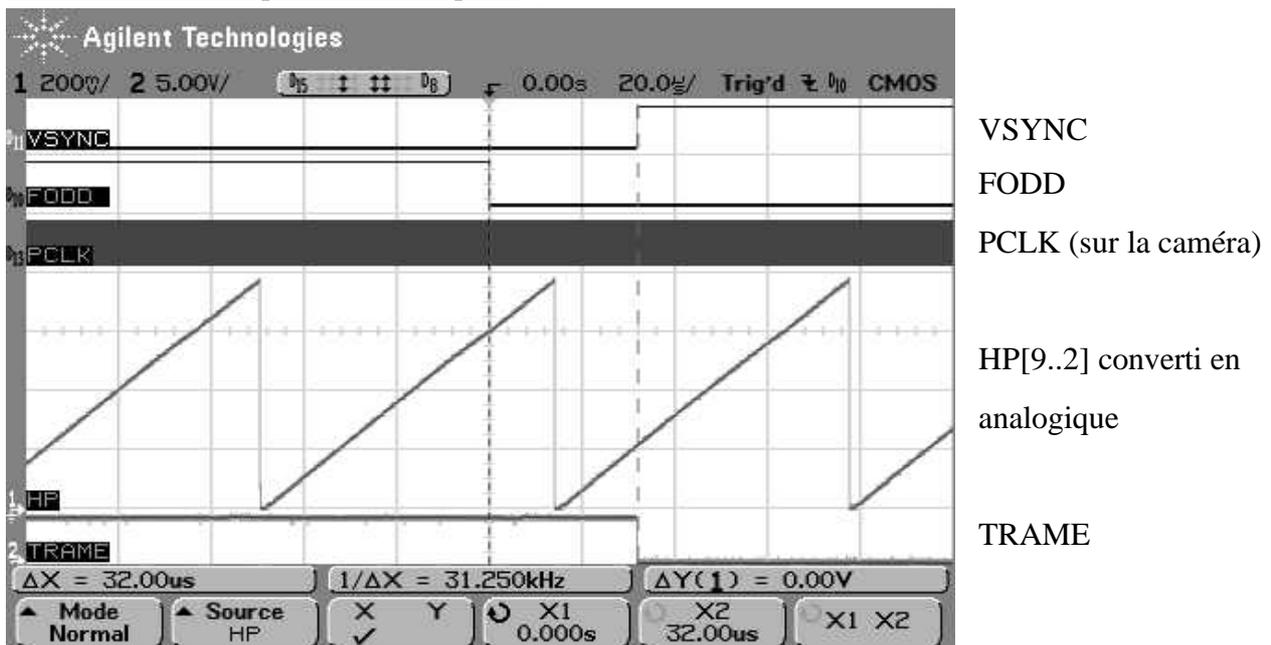
Transition trame paire → trame impaire (déclenchement sur ↑ de FODD) :



Pour visualiser graphiquement l'état du compteur HP[9..0], on applique ses 8 bits de poids fort sur le CNA "I" de la maquette d'étude. La valeur convertie en analogique varie donc de 0 à 214 quand HP[9..0] évolue entre 0 et 857.

Le signal analogique VI_OUT est appliqué sur la voie 1 de l'oscilloscope.

Transition trame impaire → trame paire (déclenchement sur ↓ de FODD) :



On constate en comparant les 2 relevés que l'état de HP[9..0] est différent dans chaque cas au moment du ↑ de VSYNC :

- HP[9..0] $\approx 3/4$ du max dans le premier cas. Soit HP[9..0] ≈ 643
- HP[9..0] $\approx 1/4$ du max dans le deuxième cas. Soit HP[9..0] ≈ 214

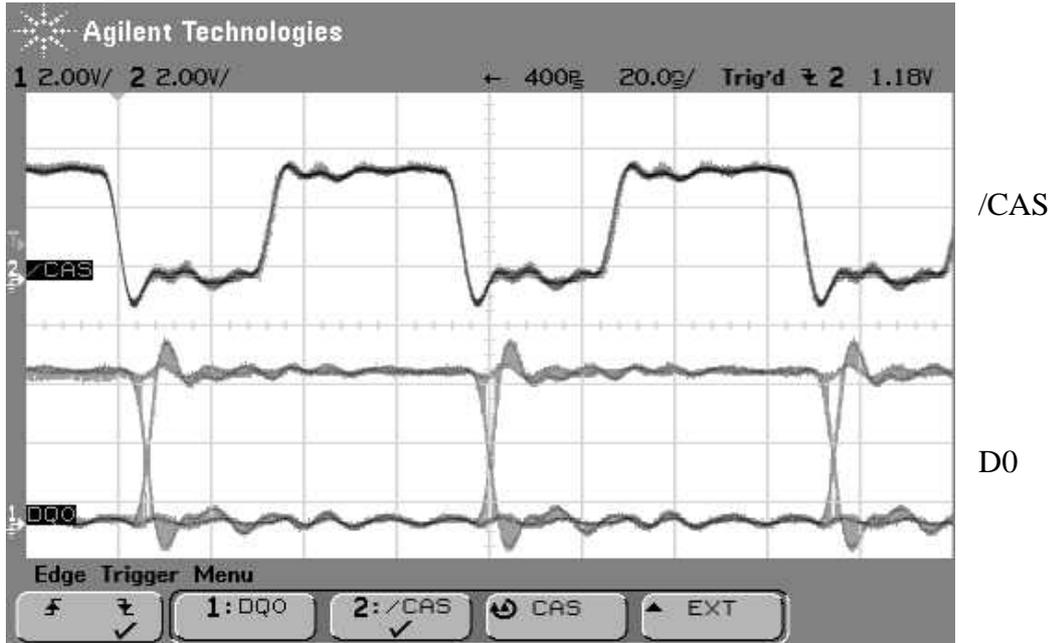
Pour identifier ces 2 cas et ainsi identifier la trame, il suffit de prendre en compte l'état du bit HP9 au ↑ de VSYNC :

- HP9 = "1" à la transition trame paire → trame impaire (car HP[9..0] $\geq 2^9 = 512$)
- HP9 = "0" à la transition trame impaire → trame paire (car HP[9..0] $< 2^9 = 512$)

Ceci est simplement réalisé par une bascule D : D=HP9 et CLK=VSYNC. Toutefois, dans la description définitive, VSYNC est d'abord échantillonnée par PCLK pour éliminer les aléas dus aux temps de commutation excessifs de VSYNC.

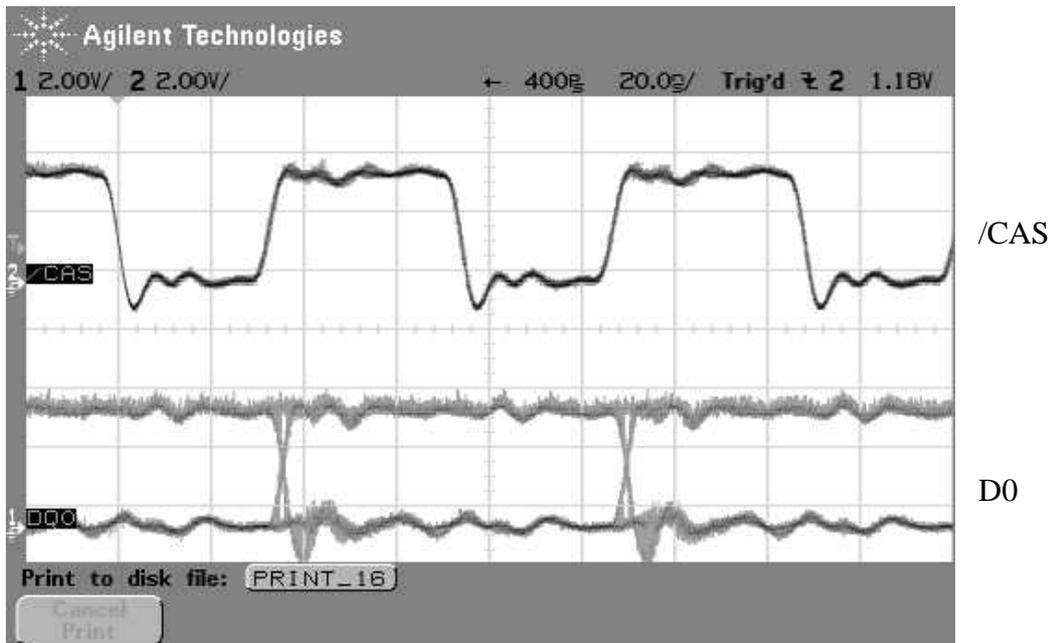
1.4 Accès données DRAM

Lecture :



La DRAM est utilisée en mode EDO et l'échantillonnage des adresses "lignes" a été effectué avec /RAS. Le signal PCLK_O (non représenté) est pratiquement identique à /CAS (déphasage négligeable). On constate un temps d'accès très court à la donnée DRAM : < 10ns ce qui donne un diagramme de l'œil très ouvert.

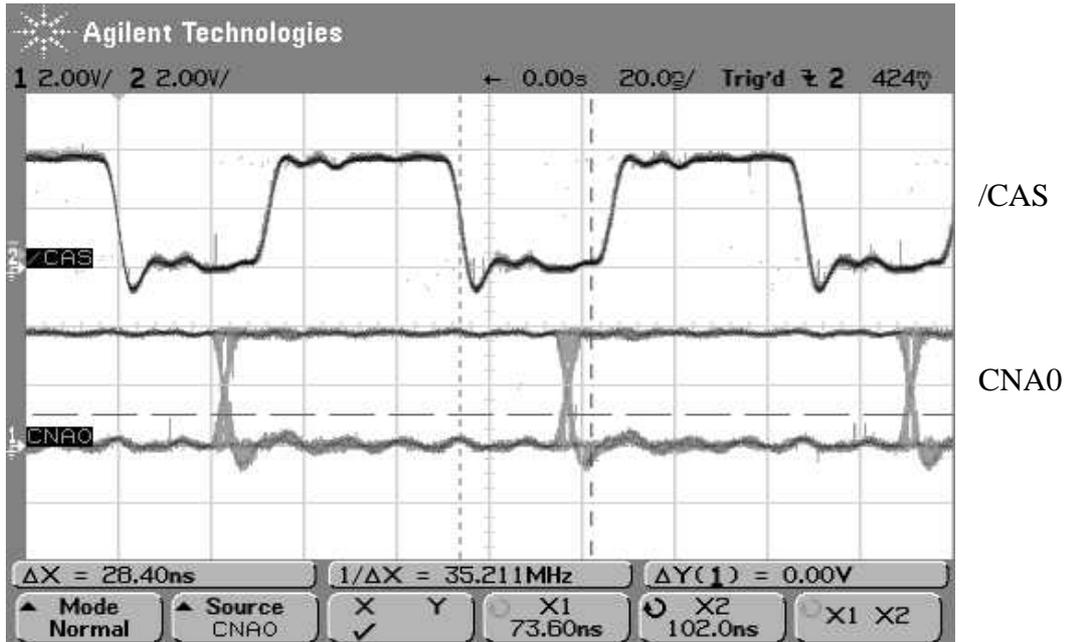
Écriture :



La documentation de la DRAM indique que la donnée à écrire doit être stable de part et d'autre du ↓ de /CAS (10ns pour les plus lentes). Ce critère est largement respecté.

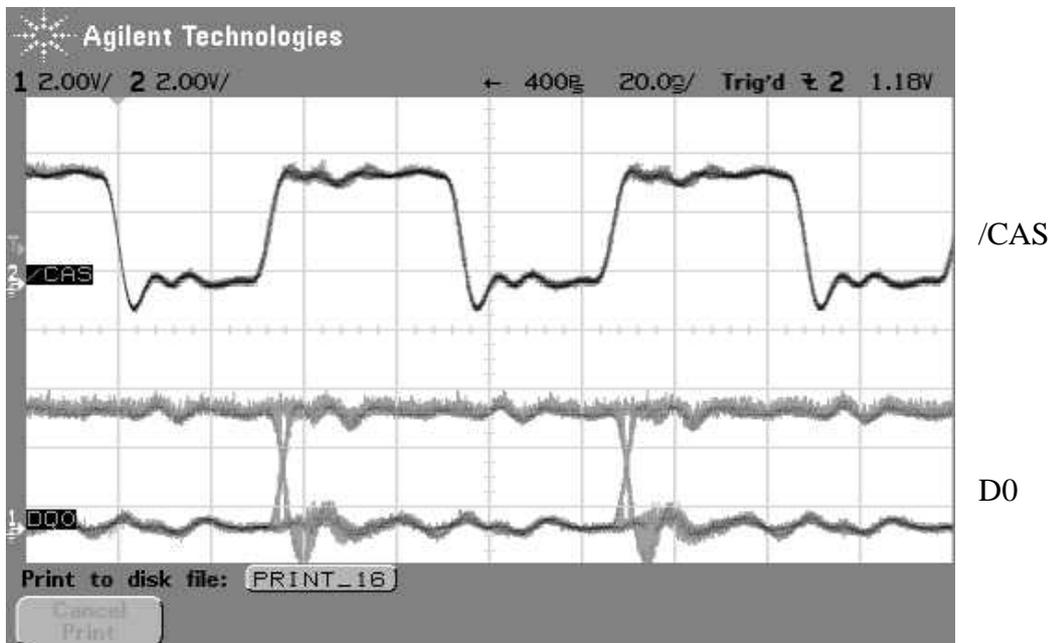
1.5 Sortie CNA

Lecture DRAM :



La fonction "CNA_out" ne comporte pas le registre tampon. Le retard supplémentaire constaté par rapport aux chronogrammes précédents est dû essentiellement à la fonction "Multiply" de "CNA_out". Le convertisseur A/N comporte un registre d'entrée actif aux flancs montants. Le signal CLK_CNA est l'inverse de PCLK_O (/CAS et PCLK_O sont pratiquement identiques), ce qui assure un échantillonnage fiable.

Écriture :

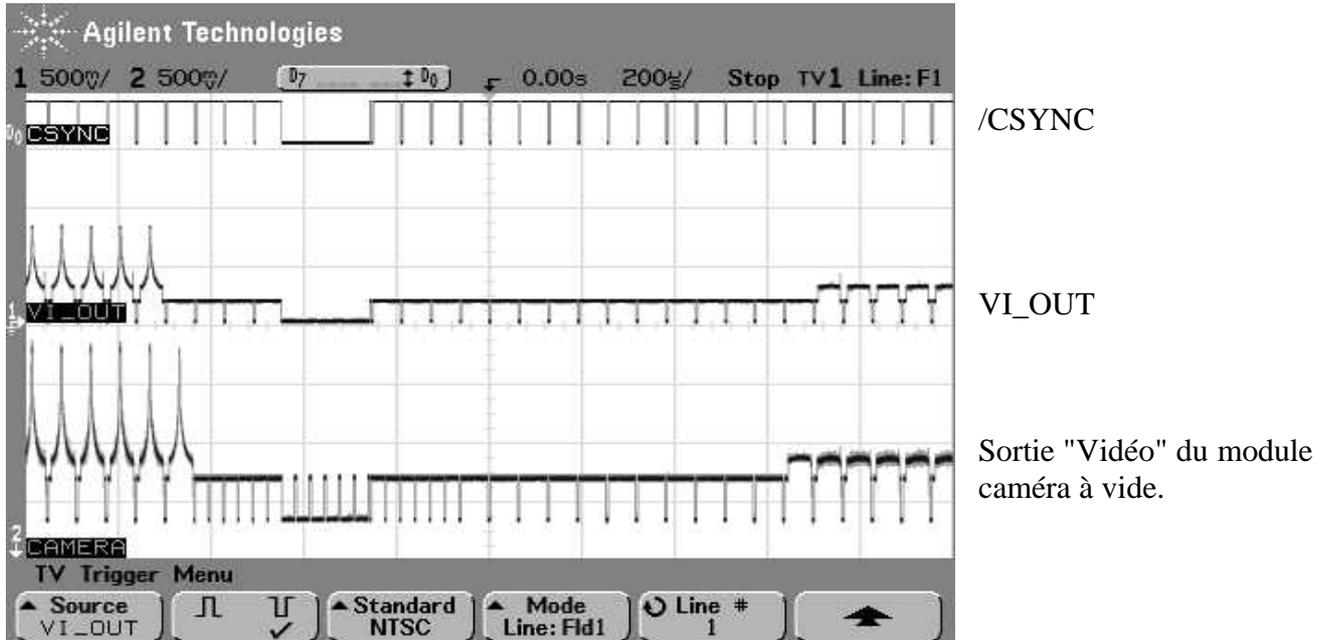


La donnée provient du registre YO[7..0] de la fonction "Driver_ZV_port_Dram" piloté par l'horloge PCLKO. Le signal CLK_CNA est l'inverse de PCLK_O (/CAS et PCLK_O sont pratiquement identiques), ce qui assure un échantillonnage fiable.

1.6 Production du signal vidéo composite

Il s'agit d'un signal N&B de contrôle de l'image. La fonction "CNA_out" combine les signaux YO[7..0], /CSYNC et EN_VID (=HREF) pour produire ce signal.

Le signal /CSYNC est "simplifié" comme le montre le chronogramme ci-dessous :



La caméra vise une ampoule électrique allumée dans le bas de l'image de façon que le signal vidéo atteigne le niveau blanc.

On constate que :

- La composante synchro du signal vidéo composite produit ne respecte pas exactement la norme, mais suffisamment pour synchroniser un moniteur vidéo.
- Les niveaux de tension de VI_OUT sont conformes
- Les niveaux de tension du signal vidéo du chip caméra ne sont pas conformes ! (un peu faibles)

A noter que le signal vidéo N&B produit par la caméra comporte une ligne supplémentaire en bas et en haut de l'image.

2. Caméra inactive : lecture DRAM depuis le PC (RST="1")

2.1 Extrait du programme DELPHI

Il s'agit de la procédure activée au déclenchement de l'acquisition :

```
procedure TForm1.LedButton_CaptureClick(Sender: TObject);
var i,h,v,y_uv : integer;
    s : string;
    Start_Time: DWORD;
begin
  New_R_W(true); {Port Lpt en lecture, par prudence}
  Start_Time:=GetTickCount;
  repeat
    if GetTickCount-Start_Time>2000 then
      begin
        Showmessage('Pas de synchro trame ou caméra non connectée !');
        exit;
      end;
  until not Test_ERROR; {Attendre FODD=ERROR="1":fin de trame}
  repeat until Test_ERROR; {Attendre FODD=ERROR="1":fin de trame}
  New_INIT(true); {INIT=RST="1" -> module C38 inhibé (Hiz)}
  New_SLCTIN(true); {VSYNC="1" -> Validation des initialisations}
  New_STROBE(true); New_STROBE(false); {Impulsion sur PCLK : Raz HP et VL=19}
  New_SLCTIN(false); {Fonctionnement normal}
  for y_uv:=0 to 1 do
    for v:=0 to Nb_pixel_V-1 do
      begin
        New_STROBE(true); New_STROBE(false); {Impulsion sur PCLK: /RAS=0}
        for h:=0 to Nb_pixel_H-2 do {Attention : 639 pixels}
          begin
            New_STROBE(true); New_STROBE(false);{Impulsion sur PCLK:/CAS=1 puis 0}
            if v < Nb_pixel_V div 2 then i:=2*v
              else i:=2*v-Nb_pixel_V+1;

            if y_uv=0
            then
              begin
                Image_Y[i,h]:=Read_DATA_LPT; {Lecture pixel}
                Image_Y[i,h]:=Read_DATA_LPT; {Lecture pixel}
              end
            else
              begin
                Image_UV[i,h]:=Read_DATA_LPT; {Lecture pixel}
                Image_UV[i,h]:=Read_DATA_LPT; {Lecture pixel}
              end
            end;
          end;
        end;
      end;
    end;
  end;
```

```

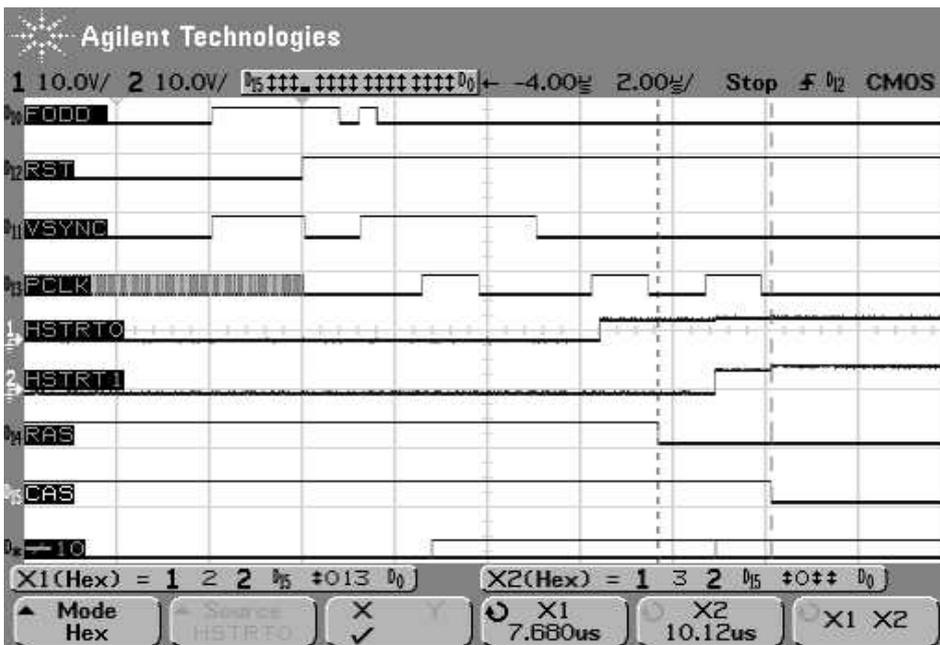
end;
end;
New_STROBE(true); New_STROBE(false); {Impulsion sur PCLK: /RAS=1}
end;
New_INIT(false); {INIT=RST="0" -> module C38 validé}
...

```

Notes :

- La procédure a été tronquée de ses dernières lignes non utiles aux commentaires des chronogrammes.
- La fonction "Test_ERROR" renvoie l'état de la ligne ERROR c'est à dire FODD (voir schémas)
- Les procédures "New_INIT", "New_SLCTIN" et "New_STROBE" affectent les lignes associées (INIT = RST, SLCTIN = VSYNC, STROBE = PCLK).

2.2 Début de la lecture



FODD
RST
VSYNC
PCLK (sur la caméra)
H_START0 (voie 1 oscillo)
H_START1 (voie 2 oscillo)
/RAS
/CAS
AM[9..0]

Acquisition déclenchée au flanc montant de RST

Note : l'horloge PCLK_O interne à l'EPLD est légèrement retardée par rapport à PCLK à cause de la fonction anti-rebonds incluse dans "Make_clock". Cela explique les petits retards de réaction des structures synchrones.

On constate que :

- RST est bien mis à "1" peu de temps après la mise à "1" de FODD conformément aux 2 boucles d'attente du début de la procédure "TForm1.LedButton_CaptureClick" : repeat ... until not Test_Error. Le temps de réaction dépend du PC, mais surtout de Windows ... !
- Jusqu'à ce moment, les signaux FODD, VSYNC et PCLK sont produits par la caméra, avec notamment un signal PCLK de fréquence 13,5MHz.
- Après ce moment, ces mêmes signaux sont produits par le PC (sauf FODD) via le port imprimante. L'état logique du signal FODD est fixé par une résistance de rappel vers 0V intégrée dans le chip caméra, ce qui explique les quelques rebonds observés dans le chronogramme.
- Le programme exécuté par le PC fait passer VSYNC à "1" puis produit une impulsion sur PCLK. Cela provoque dans l'EPLD, les initialisations synchrone sur PCLK des compteurs HP[9..0] et VL[9..0] et de H_START[1..0] : HP[9..0] ← 0, VL[9..0] ← 19, H_START[1..0] ← 0.
- Ces initialisations sont vérifiées sur le chronogramme, sauf HP car H_START0="0" ⇒ AM[9..0]=VL[9..0]
- VSYNC revient à "0" et on entre dans les 3 boucles de lecture de la DRAM : pour la luminance, puis la chrominance → pour chacune des 480 lignes utiles → lecture des 639 pixels utiles de chaque ligne.

- Chaque lecture de ligne commence par l'échantillonnage de l'adresse "lignes" dans la DRAM : la 2° impulsion PCLK provoque la mise à "1" de H_START0 puis la mise à "0" de /RAS. Au ↓ de /RAS, on relève AM[9..0]=VL[9..0]=13_H=19. C'est conforme.

Note : cette 2° impulsion n'incrmente pas le compteur HP (HP[9..0] reste à 0) car /RAS="1" au ↑ de PCLK, mais cela ne peut être relevé car H_START1="0" et AM[9..0]=VL[9..0].

- Le 1° pixel est lu à la 3° impulsion PCLK : le compteur HP s'incrmente, H_START1 passe à "1" ce qui modifie AM[9..0]=HP[9..0] puis /CAS passe à "0".

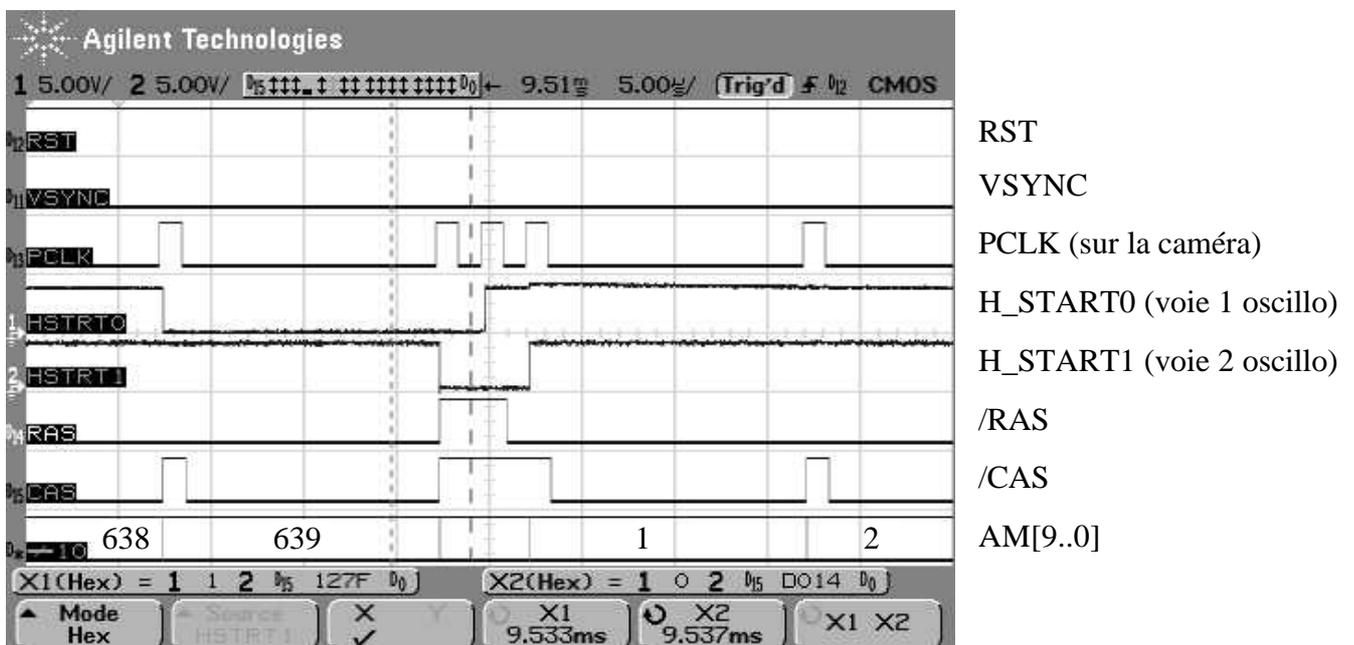
Au ↓ de /CAS, on relève AM[9..0]=HP[9..0]=1. C'est conforme.

- /RAS et /CAS sont maintenant à "0" et la DRAM fournit sa donnée. Celle-ci est aiguillée vers le port Y[7..0] de la caméra via la fonction "Mux2" et le buffer "TRI_8".

Le port Y[7..0] est alors lu par le PC via le tampon U1 du module d'interface (voir schéma).

Note : Les résistances de protection R15 avec les condensateurs de découplage CEM dans le PC augmentent les temps de commutation, ce qui nécessite un délai à la lecture réalisé par l'appel de plusieurs procédures "Read_DATA_LPT".

2.3 Passage à la ligne 20



La ligne 20 est la 2° ligne utile de la trame impaire. On veut vérifier ici que la seule activation de PCLK fait évoluer correctement les compteurs HP et VL.

Rappel : le signal H_START1 détermine AM[] : H_START1= "0" → AM[9..0]=VL[9..0]
 "1" → AM[9..0]=HP[9..0]

On constate que :

- Le compteur HP est dans l'état 638 au début du chronogramme. Il passe naturellement à 639 au flanc montant de la première impulsion PCLK représentée (curseur X1 : 27F_H=639)
- L'état 638 est décodé dans l'EPLD et provoque la mise à "0" de H_START0 au même moment.
- A l'impulsion PCLK suivante,
 - HP[9..0] passe à 0 car il compte modulo 640 (ne peut être vérifié)
 - VL[9..0] s'incrmente : passage à la ligne suivante.
 - H_START1 passe à "0" car H_START0 est à "0". Par conséquent :
 - AM[9..0]=VL[9..0]=014_H=20 (curseur X2) ce qui vérifie l'incrmentation du compteur VL
 - /RAS et /CAS reviennent "1" pour préparer la lecture de la ligne 20
- A la 3° impulsion PCLK du chronogramme :
 - H_START0 revient à "1" car HP[9..0] est à 0, ce qui provoque la mise à "0" de /RAS à la mise à "0" de PCLK
 - Au flanc descendant de /RAS : AM[9..0]=VL[9..0]=20 : l'adresse "ligne" est échantillonnée dans la DRAM

- A l'impulsion PCLK suivante :
 - HP[9..0] s'incrémente et passe donc à 1
 - H_START1 passe à "1" car H_START0 est à "1"
 - Dans ces conditions : /CAS=PCLK. On vérifie donc qu'au ↓ de /CAS : AM[9..0]=HP[9..0]=1 (pas de curseur mais vérifié à l'oscillo) : l'adresse "colonne" du 1° pixel utile est échantillonné dans la DRAM
 - /RAS et /CAS sont maintenant à "0" et la DRAM fournit sa donnée. Celle-ci est aiguillée vers le port Y[7..0] de la caméra via la fonction "Mux2" et le buffer "TRI_8" et lue par le PC.
 - Un délai assez long s'écoule jusqu'à la prochaine impulsion PCLK (environ 15µS) pour laisser se stabiliser les données.

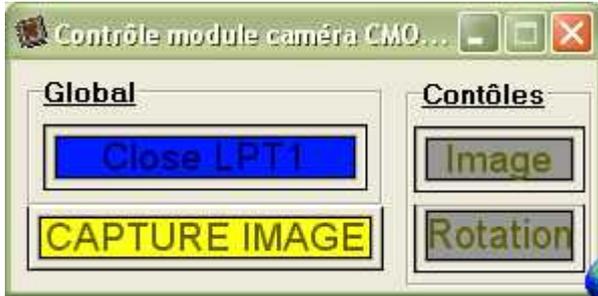
Conclusion : tout est conforme

3. Programme d'acquisition sur PC : C38A.exe

3.1 Préparation

Le module caméra C38A doit être connecté à la carte d'étude EP10K20 via le connecteur "I_O". Le module caméra est alimenté via ce même connecteur : **il ne faut donc pas l'alimenter directement !** La configuration adéquate doit être téléchargée dans l'EPLD avec MaxPlus via le port parallèle du PC. Le module caméra doit aussi être connecté au même port imprimante. Le plus pratique est d'utiliser un boîtier de partage manuel.

3.2 Lancement



La fenêtre suivante est ouverte.

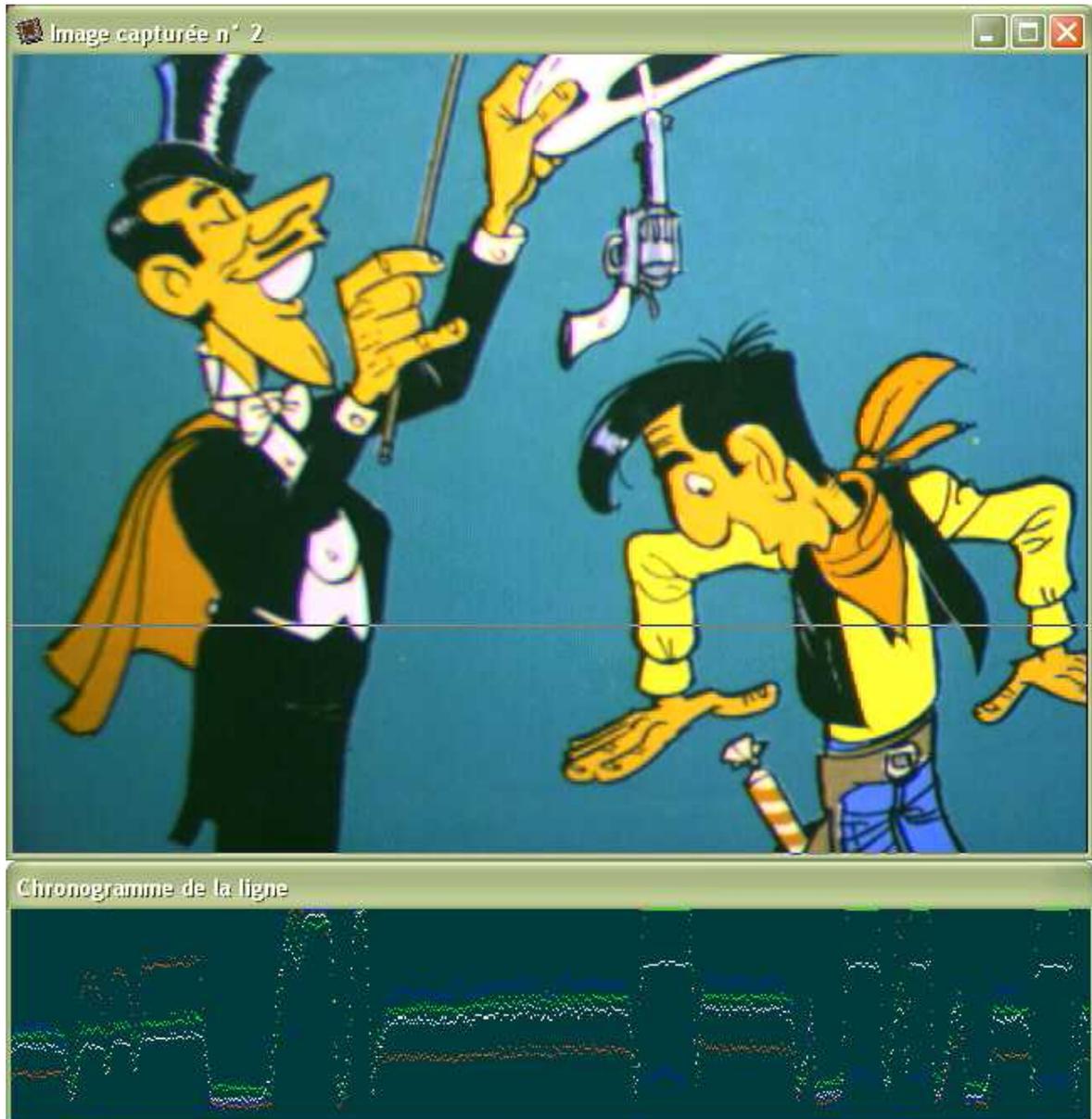
Aiguiller le port imprimante sur le module caméra. Pour capturer une image, il faut ouvrir le port LPT1 et cliquer sur le bouton "CAPTURE".

En fait, la mémorisation de l'image doit déjà être faite à ce moment là sur la maquette d'étude : appuyer sur une des touches 1, 2 ou 3 du clavier décimal.

On peut contrôler l'image N&B (en écriture et lecture)

sur l'éventuel moniteur connecté sur VQ_OUT.

3.3 Exemple d'acquisition



En bougeant la souris sur l'image, on fait apparaître une ligne en vidéo inverse dont les composantes rouge, verte et bleue sont représentées dans la feuille "chronogramme".

Dans l'exemple, on identifie facilement les zones noires (les 3 composantes proche du mini), blanches (les 3 composantes proches du max) et jaunes (composantes rouge et verte max, bleue mini).

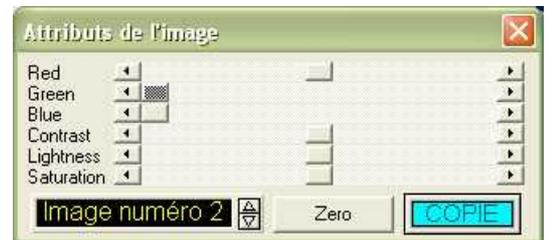
3.4 Traitements de l'image

3.4.1 Image : Composantes rouge, verte et bleue

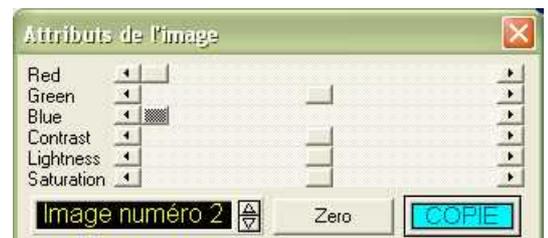
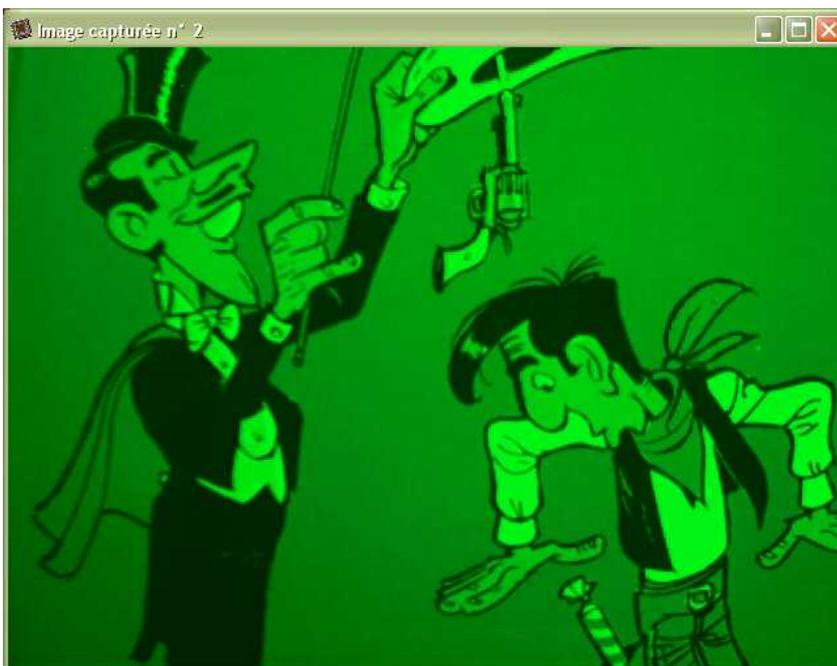


Tous les attributs sont réglables comme le montre la fenêtre ci-contre.

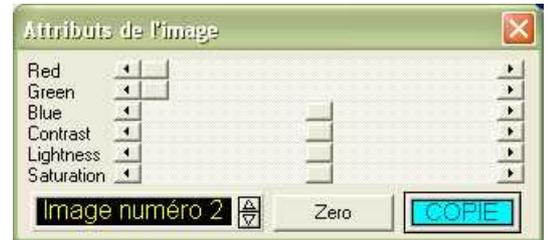
3.4.2 Exemples



Annulation des composantes verte et bleue.



Annulation des composantes rouge et bleue.



Annulation des composantes rouge et verte.

3.4.3 Rotation de l'image

On peut tourner l'image dans tous les sens !

Exemple :



3.4.4 Mémorisation de l'image

Un double-clic sur l'image ouvre la fenêtre de sélection du fichier. L'image est mémorisée au format "bmp" de 640x480 pixels.

MODULE CAMÉRA C38A

SCHÉMA STRUCTUREL & IMPLANTATION

