

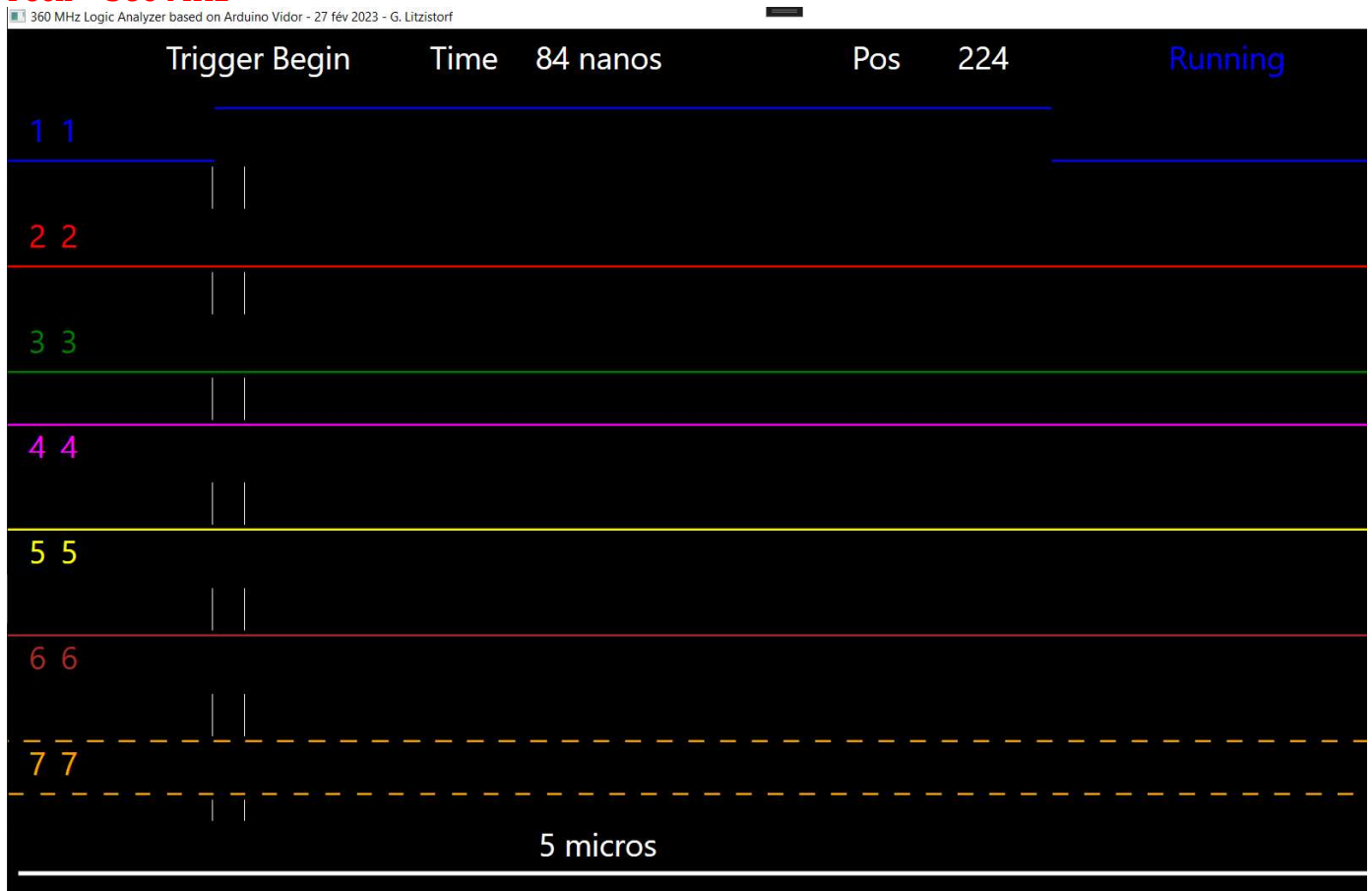
v3 du 27 fév 2023 / litzis@bluewin.ch

Principaux objectifs :

A partir du projet Analyseur Logique décrit dans <https://gelit.ch/Vidor/LA.pdf> :

- Exploiter tout le potentiel fréquentiel de cette FPGA en permettant une **fréquence d'échantillonnage F_{ech} maximale de 360 MHz**
- Utiliser une PLL en synthétiseur de fréquence
- Fournir <https://github.com/gelit/Logic-Analyzer-360MHz-with-Arduino-Vidor>

F_{ech} = 360 MHz



Les signaux sont produits avec une carte Arduino Due
Voir source dans le dossier Due

Le canal 1 correspond à `digitalWrite(2,1); digitalWrite(2,0);`

Le canal 7 reçoit un signal périodique de fréquence = 12 MHz
L'intervalle de temps mesuré entre les 2 curseurs = 84 ns

J'ai eu quelques difficultés pour travailler à 360 MHz

Heureusement que je me suis rappelé

<https://moodle.luniversitenumérique.fr/mod/page/view.php?id=1674> afin d'utiliser Global Clock Network

Utilisation

Les flèches Up & Down modifie la fréquence d'échantillonnage

Div	X	pin D7	pin D7	x 1024	pin D6 Tech 2.95 ms	pin D6 Fech	Ecran de 1900 pixel 5603 ms
...							
6	5				180 ns	5.55	
5	4	91		11	90 ns	11.1 MHz	171 µs Default
4	3	45.5		22			86
3	2	22.8		45			43
2	1	11.4		90			21
1		5.69		180			11
0		2.84 µs	0.352 MHz	360 MHz			5 µs

Avec Tech = 5.76 µs, l'impulsion du canal 1 de 2.2 µs est détectée de façon aléatoire ; ce qui est logique !

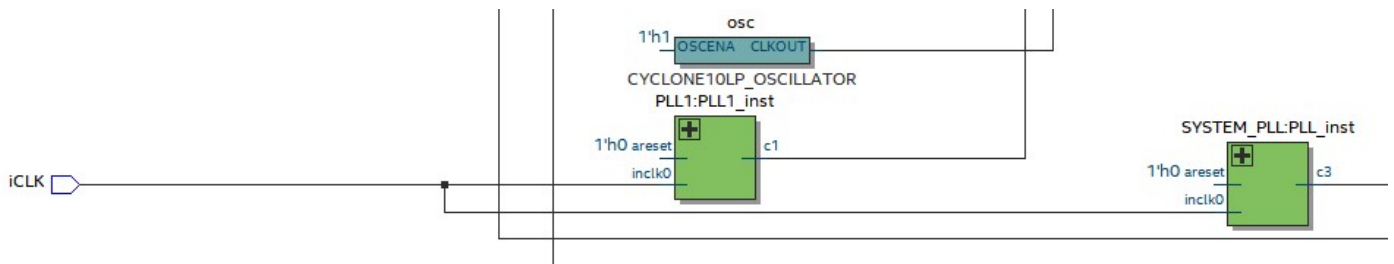
PLL en synthétiseur de fréquence

Lors de mes études d'ingénieur, j'ai eu la chance d'effectuer un travail pratique (vers 1976) dans lequel je devais configurer un synthétiseur de fréquence à partir de circuits intégrés.

Intel met à disposition 4 composants PLL dans cette FPGA Cyclone LP10

Le design de PLL1 (basé sur celui de SYSTEM_PLL) configure la sortie c1 pour une

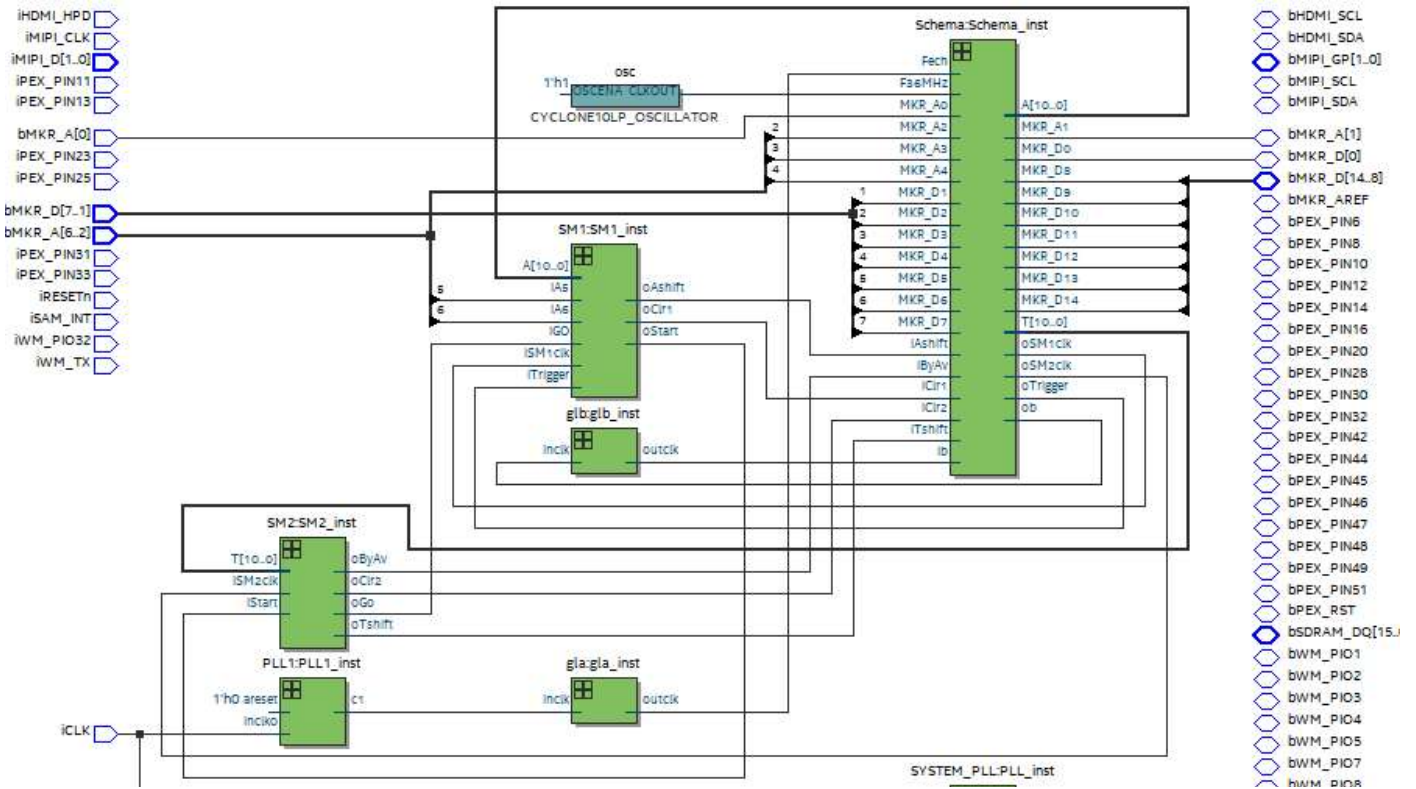
Fréquence c1 = $15 / 2 \times 48 \text{ MHz} = 360 \text{ MHz}$ tout en respectant les limites (voir Points d'amélioration dans <https://gelit.ch/Vidor/LA.pdf>)



Je n'ai pas réussi à connecter CYCLONE10LP_OSCILLATOR à la place de iCLK
 Error (15065): Clock input port inclk[0] of PLL must be driven by a non-inverted input pin or another PLL, optionally through a Clock Control block Info

Cette erreur a certainement conduit Dario Pennisi (auteur de _top.v) à utiliser iCLK = 48 MHz dans son design de SYSTEM_PLL

Vue d'ensemble avec Tools - Netlist Viewers - RTL Viewer

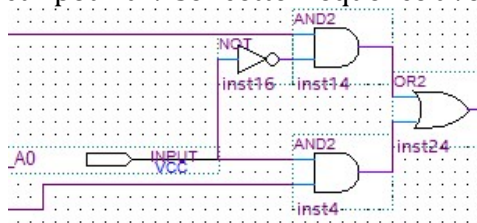


L'horloge de 360 MHz est produite par PLL1

Elle reçoit un chemin privilégié grâce à gla → https://cdrdv2-public.intel.com/654442/ug_altclock.pdf

Si pin A0 = 1 → les 360 MHz sont transmis

Sinon → Counter inst1 est actif pour diviser cette fréquence avec une puissance de 2



La sortie de la porte OU est relié à un second Clock Control Block

Après compilation, Quartus met en rouge **Timing Analyzer**

... il y a donc certainement des ajustements possibles

A propos du support HDMI de la carte Vidor

J'ai acquis 2 cartes dans le but de me familiariser avec le développement matériel basé sur une FPGA

J'avais aussi envie de tester les performances des bibliothèques HDMI disponibles
Voir page 8 de <https://gelit.ch/Vidor/LA.pdf>

Hélas, les résultats ont été très décevants avec ces 2 bibliothèques :

<https://support.arduino.cc/hc/en-us/articles/360016912700-How-to-show-video-on-HDMI-with-MKR-4000-Vidor>
<https://systemes-embarques.fr/wp/archives/sortie-video-et-son-sur-hdmi-avec-la-mkr-vidor-4000/>

Utiliser en 2023 un connecteur HDMI pour n'obtenir qu'une résolution VGA de 640 x 480 ne présente aucun intérêt puisque j'utilisais cette résolution en 1987 avec des cartes Vega

Ce lien <https://forum.arduino.cc/t/maximum-hdmi-signal-resolution/600454> confirme l'impossibilité pour cette carte de supporter une résolution de 1920 x 1080 à 50Hz

Je prévois pour 2023 d'investiguer les réelles possibilités graphiques des 2 micro connecteurs HDMI présents sur la carte <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>