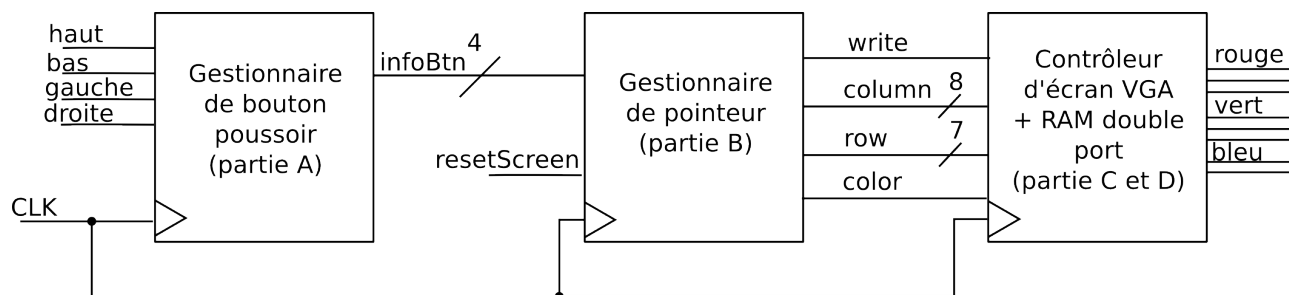


But du projet



Il s'agit de réaliser le télécran présenté en examen en 2012. Ce télécran sera piloté soit par quatre interrupteurs (v1 : joystick), soit par deux codeurs incrémentaux permettant de jouer sur les pointeurs horizontaux et verticaux (v2).

La résolution visée est de 160x120 pixels, monochrome (noir / couleur à définir, codé sur 1 bit).

Matériel nécessaire

Bibliographie :

- Sujet d'examen 2012 (indispensable ! → moodle).
- Le « Reference Manual » de la carte Basys 2(→ moodle).
- Spartan 3 FPGA User Guide (pp. 153 à 186 : section sur la RAM par bloc → moodle).
- Le Data Sheet, au choix des codeurs incrémentaux, ou bien d'un joystick (→ moodle).

Matériel :

- Carte Basys2 avec horloge stabilisée 25MHz.
- 1 écran LCD VGA 17".
- (selon la version) 2 codeurs incrémentaux (v1) ou un joystick (v2).

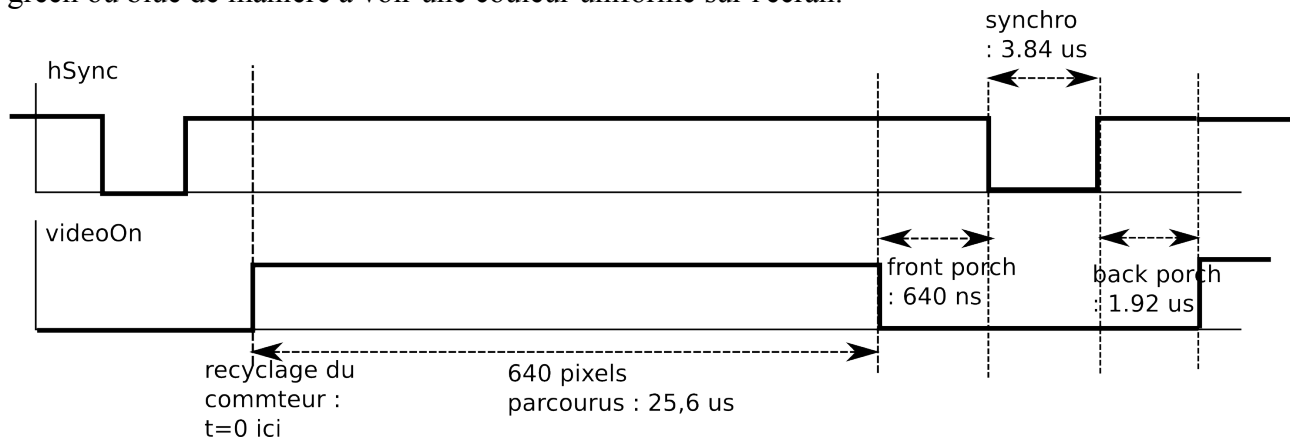
Remarque importante : l'horloge de la carte, stabilisée, est à 25 MHz, afin d'obtenir un affichage sans défaut. Il faut donc revoir les calculs du sujet à cet égard.

Première étape (VHDL)

Dans ce projet, le mieux est de commencer par la fin. Codez la génération de synchronisation. Un exemple est fourni dans le corps du sujet. Si cela ne fonctionnait pas (temps de stabilité dans les bascules), vous pouvez le coder avec deux bascules RS actionnées par des comparateurs (cf RM Basys). Vous devez également sortir les informations de numéro de ligne et de colonne (numéro de ligne : doit varier entre 0 et 119, numéro de colonne : doit varier entre 0 et 160, on peut faire une division par 4 en enlevant deux lsb), ainsi que le signal « vidéoOn ».

Testez le sur maquette. Attention au respect des temps. Si les synchro horizontale et verticale ne sont pas assez précise (1%), vous risquez de ne rien voir par la suite. Prenez le temps de valider complètement le fonctionnement. Pour ce faire, dupliquez le signal VideoOn sur une sortie red,

green ou blue de manière à voir une couleur uniforme sur l'écran.



Deuxième étape (Un peu de lecture... puis du VHDL !)

Creez un composant intégrant deux RAM 1 bits (14 bits d'adresse) capable de prendre en entrée l'ensemble lineNumber et colNumber et d'y inscrire un bit ou d'y lire un bit. ATTENTION: il ne faut pas coder une RAM en VHDL, sinon le compilateur va essayer de la mettre dans la partie programmable du circuit. Il faut activer par une syntaxe spéciale (une primitive) la RAM préintégré sur le composant. Pour cela, on lira avec profit le Spartan 3 user guide sur Moodle.

Le méta composant sortira 8 bits valant soit une entrée (par exemple les interrupteurs) si la valeur lue en RAM vaut '0' soit « 00 » si la valeur lue en RAM vaut '1'. Vous devez également tenir compte du signal videoOn.

Intégrez l'ensemble des deux premières parties. Normalement, vous devez afficher une couleur uniforme sur toute la surface de l'écran.

Troisième étape (Du code, toujours)

Coder les compteurs saturés au centre de la partie B. Le gros séquenceur signalé dans le sujet d'examen n'est pas nécessaire tant qu'on ne gère pas le reset (voir partie : cerise sur le gâteau). InfoBtn fourni directement EnableRUp, EnableRDown, EnableCUp, EnableCDown. Write vaut '1' et Color vaut '1' en permanence (utilise uniquement si l'on veut effacer l'écran).

Quatrième étape (enfin du matériel !)

Vous devez désormais choisir entre :

- Un télécran à bouton (sujet d'examen). Les boutons seront réalisés par un joystick fournis par votre encadrant de mini-projet.

Un télécran à codeur rotatif (plus rigolo, mais il faut s'éloigner du sujet).

Petite carte à créer : comportant l'un ou l'autre des dispositifs, il ne reste qu'à prévoir l'interconnexion avec les PMODs, les résistances de Pull-up, et les filtres anti-rebonds.

Cinquième étape (la fin)

Dans la v1, le séquenceur du sujet d'examen est tout à fait fonctionnel, à recopier en 4 exemplaires pour chaque interrupteur. Dans la v2, il faut prévoir un autre séquenceur permettant de déterminer l'avancement et le sens de rotation du codeur. Il pourrait ressembler à celui page suivante. Il faut lire la documentation du codeur pour comprendre le séquençage.

La cerise sur le gâteau

Un peu plus difficile : codez le reset de l'écran. Vous pouvez également proposer une mémorisation dans les deux blocs RAM restant de l'image dessinée, et un rappel de cette image. On peut également envisager que le pointeur soit d'une deuxième couleur assez aisément.

Séquenceur de gestion des codeurs incrémentaux. P1 : $A=B=0$, P2 : $A=1, B=0$, P3 : $A=B=1$...

