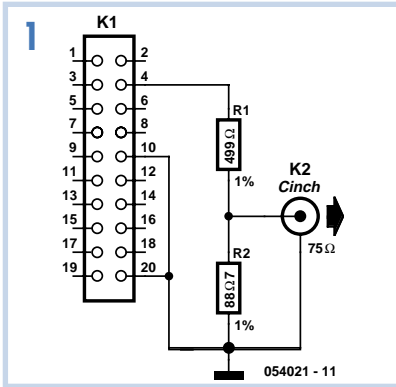


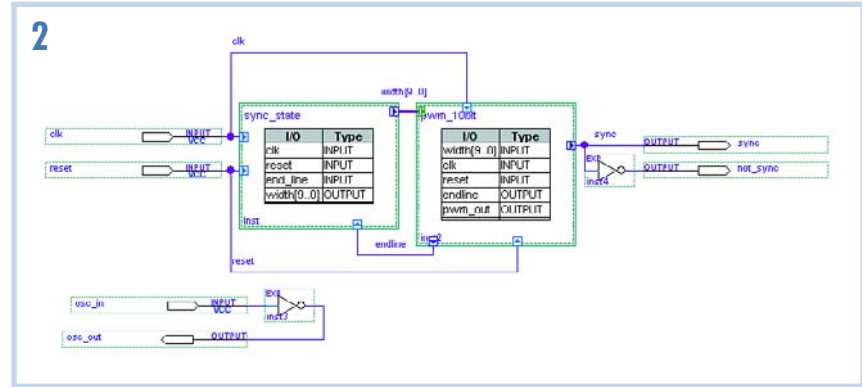
Générateur de synchronisation vidéo



Un signal vidéo contient des signaux de synchronisation. Ceux-ci sont utilisés pour permettre de synchroniser le balayage vertical et le balayage horizontal avec l'image. Il peut être intéressant pour l'expérimentateur de disposer d'un générateur de tels signaux de synchronisation.

La conception de ces signaux de synchronisation vidéo est assez compliquée et la réalisation d'un générateur analogique assez précis poserait bien des problèmes. Un CPLD nous facilitera en revanche les choses.

Nous ressortons pour ce projet une platine d'expérimentation présentée en mai 2004 dans Elektor ('Forgez votre propre puce'). La **figure 1** présente l'extension matérielle demandée par notre générateur. Comme vous le voyez, cette extension brille par sa simplicité. Le connecteur à 20 contacts est relié par un câble en nappe au connecteur K3 de la platine d'expérimentation. Les signaux de synchronisation de la broche 4 sont appliqués au connecteur de sortie par l'intermédiaire du diviseur de tension R1/R2. Deux tâches sont confiées aux résistances R1 et R2: elles garantissent d'abord une impédance de sortie de 75 Ω , elles adaptent ensuite l'amplitude du signal de quelque 4 V à 0,6 V. Si le montage est connecté à l'entrée d'un téléviseur dont l'impédance est de 75 Ω , sa tension de sortie ne doit pas dépasser 0,3 V pour



satisfaire aux conditions imposées aux signaux CVBS.

Vous l'avez déjà deviné, la plus grande partie du projet est logée dans le CPLD. Nous avons conçu ce projet en Quartus 4.2. La **figure 2** en montre le schéma. Les trois parties du projet y sont nettement identifiables. L'inverseur, en bas, fournit, avec le quartz du circuit imprimé, le signal d'horloge. Il est indispensable pour ce circuit que le quartz stabilise la fréquence à 10 MHz. Si un autre quartz est implanté, vous devez d'abord le remplacer par un quartz de 10 MHz.

Ce signal d'horloge est réinjecté au circuit par une autre entrée, sur le côté supérieur. Il garantit la synchronisation des deux parties du projet.

La partie appelée 'sync-state' détermine l'ordre dans lequel se succèdent les parties du signal vidéo. Nous devons dire ici que, pour ce circuit, chaque ligne vidéo est divisée en deux parties. Il en est ainsi à cause de la fréquence double du signal de synchronisation pendant le retour vertical.

La sortie 'width' (largeur) définit la durée pendant laquelle le signal de sortie doit être maintenu à l'état bas. Pour cette partie du circuit, le passage à l'état actif de l'entrée 'endline' (fin de ligne) indique la fin de cette partie du signal vidéo. C'est donc au tour de la partie suivante et le bloc 'sync_state' (état de synchronisation) produit la valeur correspondante à la sortie.

Le nom donné au dernier bloc 'pwm_10bit' est assez éloquent. Comme vous le supposez, cette partie (une partie du programme en fait) produit une impulsion dont la longueur est égale à la valeur 'width'. Cette modulation de largeur d'impulsion (PWM) compte de 0 à 319, ce qui, pour un signal d'horloge de 10 MHz correspond à une durée totale de 32 μ s (la moitié d'une ligne vidéo), divisée elle-même en 320 intervalles de 0,1 μ s. Pendant l'avant dernier intervalle, la PWM fait passer la sortie 'endline' à l'état haut. Nous devons, à vrai dire, veiller à ce que ce signal s'active pendant le dernier intervalle de la période. En activant cette sortie un petit peu plus tôt, nous laissons cependant au bloc 'sync-state' au moins un cycle d'horloge pour produire une nouvelle valeur de 'width'.

Notre programme pour le circuit est évidemment disponible sur le site (www.elektor.fr, EPS 054021-1 1). Le résultat définitif, après compilation, occupe 65 macrocellules. Une optimisation permettrait de réduire le nombre de cellules nécessaires. Des améliorations possibles de l'efficacité du programme sont évidentes, sur certains points, d'autres demandent un peu plus de réflexion. Essayez de voir jusqu'à quel point vous pouvez améliorer notre projet. Le sujet a de quoi animer le forum !