

Projet de conception    Lecteur vidéo

Durée                      5 semaines

---

## 1. Objectif

Comprendre les principes de conception des logiciels dédiés et multi tâches, utilisant des threads et des processus. Le laboratoire permettra d'expérimenter avec quelques mécanismes de communications.

## 2. Description

Votre mandat est de concevoir, coder, debugger et tester un logiciel permettant :

- D'ouvrir et de fermer des fichiers vidéo.
- De contrôler la lecture avec les commandes habituelles : lecture, arrêt, pause, avance rapide, recul rapide.
- Le logiciel doit inclure les traitements suivants, appliqués dans le même ordre : brillance, contraste et convolution 3x3.
- Afficher la vidéo et, possiblement, afficher plusieurs vidéos à différentes étapes de traitement.
- Le logiciel doit également permettre l'enregistrement d'un segment de la vidéo sur disque en temps réel.
- Le logiciel doit fonctionner même si un autre programme exige toute la capacité du CPU.
- Le logiciel doit être convivial, ne doit jamais bloquer ou laisser l'utilisateur attendre plus de 0.5 sec, en cas d'attente indiquer le temps restant ou un sablier.

Afin de mettre en application les principes des applications multi threads/processus :

- Vous devez avoir au moins deux processus, dont un pour l'affichage.
- Un des processus doit utiliser plus d'un thread. Par exemple, un pour la console et un pour la lecture de la trame et un autre pour les traitements.
- Vous devez utiliser des variables partagées pour les communications entre les threads et au moins un mécanisme de synchronisation.
- Pour terminer, vous devez transmettre l'image au processus d'affichage par MMF Memory Mapped File.

## 3. Déroulement

Le laboratoire se divise en trois parties :

- A. Préparer la spécification, maximum 10 pages, tout inclus (suivre le guide et utiliser le gabarit), dû jeudi le 12 novembre.
- B. Concevoir, coder, debugger et tester l'application. À cette étape vous devez fournir une architecture et une courte description de l'architecture, des modes de communications et du fonctionnement.
- C. Intégrer un traitement sur FPGA.

## 4. Évaluation

- Architecture et définition des interfaces, des modes de communications et des messages 30%.
- Qualité du code source 30%.
- Compilation et fonctionnement sans problème 40%.