

Membres du groupe : Guillaume Leinen, Jean-Noël Clink, Hussein Saad, Alexandre Froehlich, Hugo Questroy

Encadrants : Pascal Cotret, Jean-Christophe Le Lann, Joël Champeau

Code disponible sur GitHub ==> <https://github.com/EVEEX-Project>

Eportfolio Mahara disponible => <https://mahara.ensta-bretagne.fr/view/groupviews.php?group=348>

Abstract

Most of the world's bandwidth is nowadays used for the exchange of multimedia content, especially video streaming. With more than 2 billion users every month on Youtube alone [1], video streaming has not only become a taken-for-granted technology, but one that consumes a lot of resources both in terms of energy and storage.

To reduce these excessive consumptions, a solution is video compression. While business consortia compete for royalties on their proprietary compression algorithms such as MPEG-1 or x264, we want to offer a viable and open source alternative.

During this school year, we are going to design an algorithm to compress a raw video stream to send it over a network or simply to store it in a file, and then a second algorithm to decompress this data. Once the algorithm is operational and mature, we will carry out a purely hardware implementation in order to optimise the computing time and energy consumption related to data compression.

We will use several programming languages to build different prototypes incrementally. We are going to start with Python to create a naive algorithm that everyone can understand, then we are going to translate it into C to optimise its execution and above all to start getting closer to a hardware implementation. Finally we will translate this last code using LiTex to create the hardware implementation on FPGA.

It is obvious that to carry out these steps we cannot rely on existing software libraries for reasons of code portability to different languages and then to different platforms.

The objective at the end of this school year is to obtain a marketable product for video stream compression and transmission over the network using a webcam and two FPGA cards, all of course open source and available on GitHub.

Résumé

La majeure partie de la bande passante mondiale est aujourd'hui utilisée pour l'échange de contenu multimédia, en particulier le streaming vidéo. Avec plus de 2 milliards d'utilisateurs chaque mois sur Youtube seulement [1], le streaming vidéo est devenu une technologie considérée comme prise pour acquise, mais qui consomme beaucoup de ressources tant en termes d'énergie que de stockage.

Pour réduire ces consommations excessives, une solution est la compression vidéo. Alors que les consortiums d'entreprises se font concurrence pour les redevances sur leurs algorithmes de compression propriétaires tels que MPEG-1 ou x264, nous voulons offrir une alternative viable et open source.

Au cours de cette année scolaire, nous allons concevoir un algorithme pour compresser un flux vidéo brut afin de l'envoyer sur un réseau ou simplement de le stocker dans un fichier, puis un second algorithme pour décompresser ces données. Une fois que l'algorithme sera opérationnel et mature, nous procéderons à une implémentation purement matérielle afin d'optimiser le temps de calcul et la consommation d'énergie liés à la compression des données.

Nous utiliserons plusieurs langages de programmation pour construire différents prototypes de façon incrémentale. Nous allons commencer par Python pour créer un algorithme naïf que tout le monde peut comprendre, puis nous allons le traduire en C pour optimiser son exécution et surtout pour commencer à nous rapprocher d'une implémentation matérielle. Enfin, nous allons traduire ce dernier code en utilisant LiteX pour créer l'implémentation matérielle sur FPGA.

Il est évident que pour réaliser ces étapes, nous ne pouvons pas nous appuyer sur les bibliothèques logicielles existantes pour des raisons de portabilité du code vers différents langages puis vers différentes plates-formes.

L'objectif à la fin de cette année scolaire est d'obtenir un produit commercialisable pour la compression et la transmission de flux vidéo sur le réseau en utilisant une webcam et deux cartes FPGA, toutes bien sûr open source et disponibles sur GitHub.