Projet SoC : Architecture de traitement vidéo

Groupe CEFF : Cédric Honnet, Enzo Casasola, Francesco Macca, Fernando Jeronimo

2 juillet 2010

Résumé

De nos jours, l'omniprésence de flux vidéo est évidente. Évidemment, le besoin de transformation de ces vidéos s'est rapidement fait ressentir. Par exemple, des opérations basiques comme le zoom, la translation et la rotation sont particulièrement fréquents.

Dans ce contexte, la transformation *backward warping* gagne en importance. Elle utilise la position dans l'image de sortie comme entrée d'un polynôme pour obtenir en sortie la position dans l'image originale, connu comme antécédent. Au degré 1, il est déjà possible de faire en logiciel les transformations simples mentionnées précédemment. Ce projet consiste en une implémentation rapide, donc électronique, de ce procédé au 3ème degré pour plus de souplesse.

1 Introduction au Système

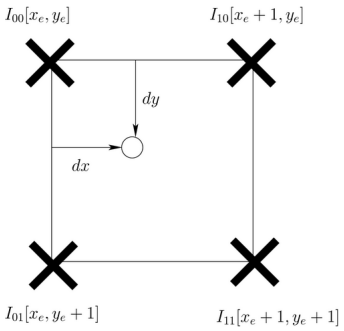
1.1 Problématique et Théorie

Voici un bref rappel du fonctionnement du *backward warping*. Soit (X,Y), les coordonnées dans image de sortie, les antécédents sont donc :

$x = a_{30}X^3 + a_{21}X^2Y + a_{12}XY^2 + a_{03}Y^3 + a_{20}X^2 + a_{11}XY + a_{02}Y^2 + a_{10}X + a_{01}Y + a_{00}$

$y = b_{30}X^3 + b_{21}X^2Y + b_{12}XY^2 + b_{03}Y^3 + b_{20}X^2 + b_{11}XY + b_{02}Y^2 + b_{10}X + b_{01}Y + b_{00}$

Dans certains cas la position (x,y) sort de l'image et le système considère le pixel transformé, celui de position (X,Y), comme étant noir. Une autre possibilité a lieu quand les parties fractionnelles de x et y, respectivement dx et dy, sont différentes de zéro, c'est-à-dire, quand l'antécédent est au milieu d'autres pixels (car les position d'image sont discrètes).
Pour ne pas perdre en qualité, l'interpolation peut être faite :



I est l'intensité du pixel :

$I = (1 - dx).(1 - dy).I_{00} + dx.(1 - dy).I_{01} + (1 - dx).dy.I_{10} + dx.dy.I_{11}$

1.2 Cahier des charges

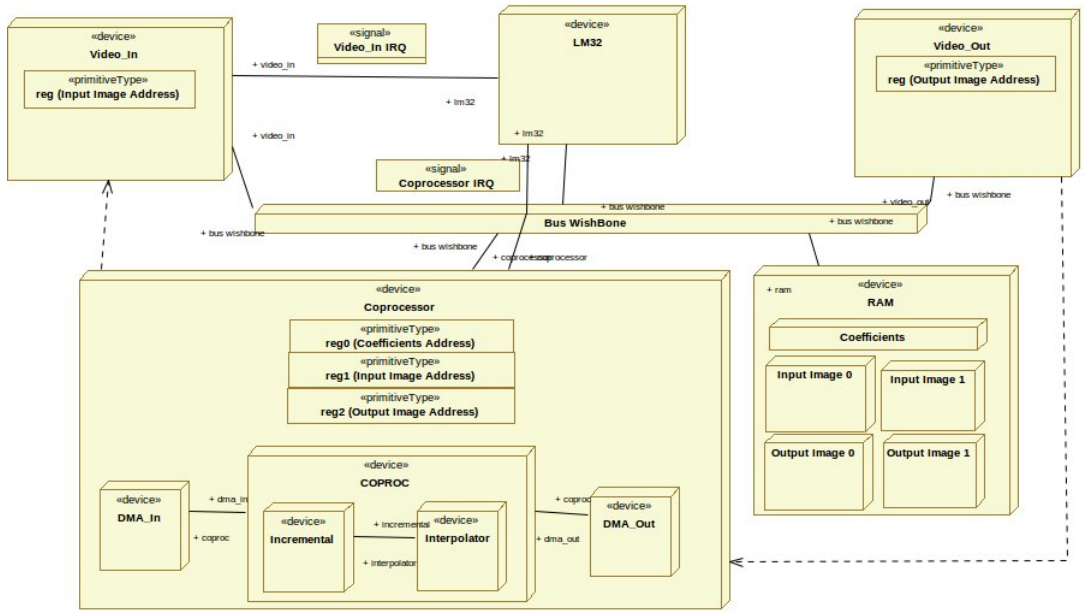
Un flux vidéo à 60 trames par seconde doit recevoir une transformation au vol qui peut se permettre un retard fixe au début. Le but est la conception d'un système ASIC permettant ces transformations. De plus, le choix du processeur est fixe, le Lattice Mico 32. L'objectif est donc d'établir et d'implémenter les parties matérielles et logicielles de façon à maximiser les souplesse et vitesse avec des coûts liés à surface, consommation et développement raisonnables.

Plus précisément, voici nos contraintes :
- Flux vidéo à 60 trames par seconde
- Résolution de 640x480 (largeur x hauteur)
- Chaque pixel est représenté par un niveau de gris dans l'intervalle [0 :255]
- La transformation doit être faite au vol
- Un retard de quelques trames est toléré
- Le processeur est le LM32
  - Fréquence maximum de 100 MHz
  - 6 étages de pipe-line
  - pas de FPU
  - 32 lignes d'interruption
- SDRAM à 100 MHz
- Bus WishBone
- Langages :
  - haut niveau - SystemC
  - processeur - C
  - RTL - SystemVerilog
- Maintenabilité

2 Architecture

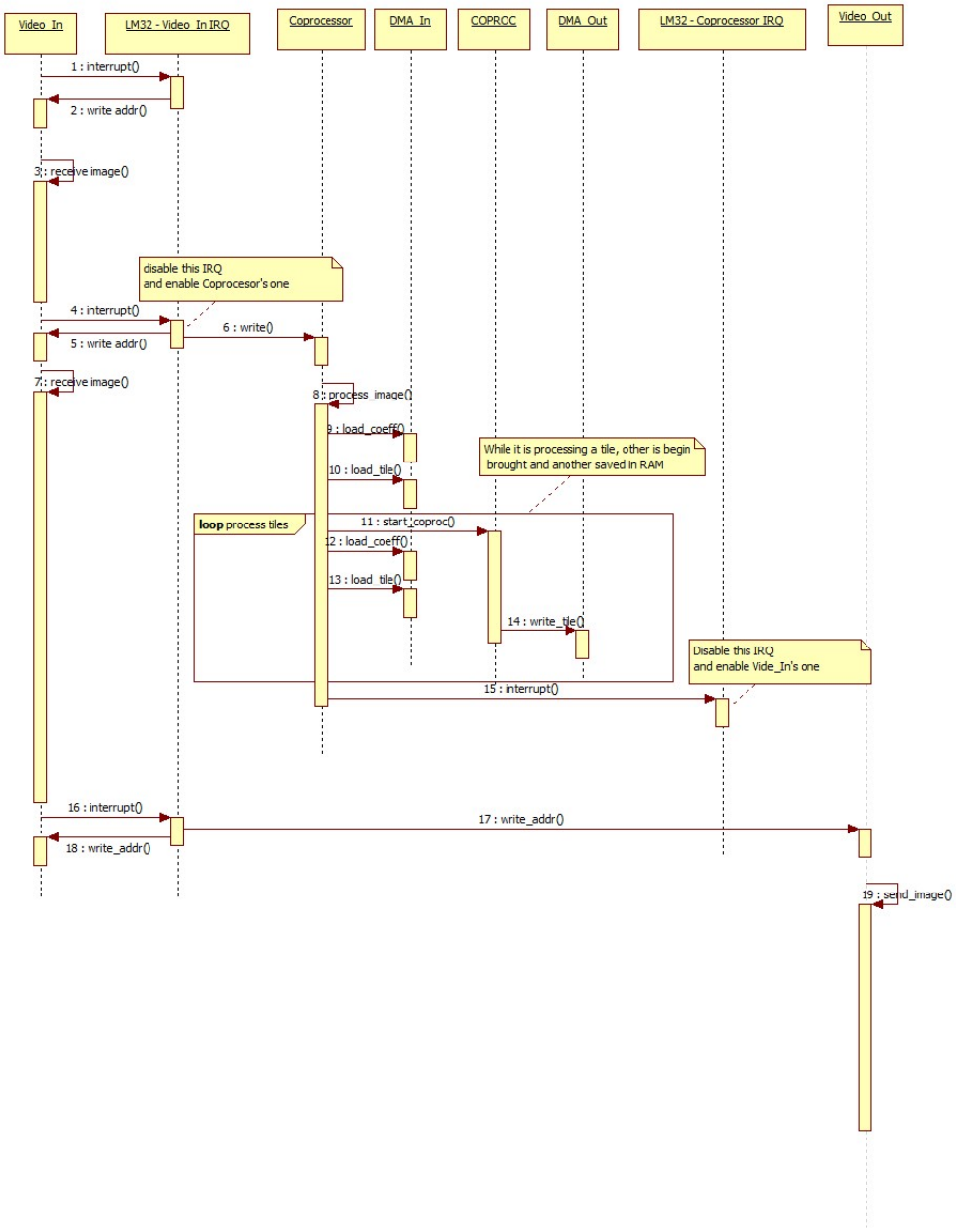
Dans cette section nous détaillons notre système à l'aide de diagrammes de structure, de séquence, ou d'état. Nous montrerons par la suite la fonctionnalité de celui-ci à l'aide de chronogramme et d'images transformées.

2.1 Structure - Deployment Diagram

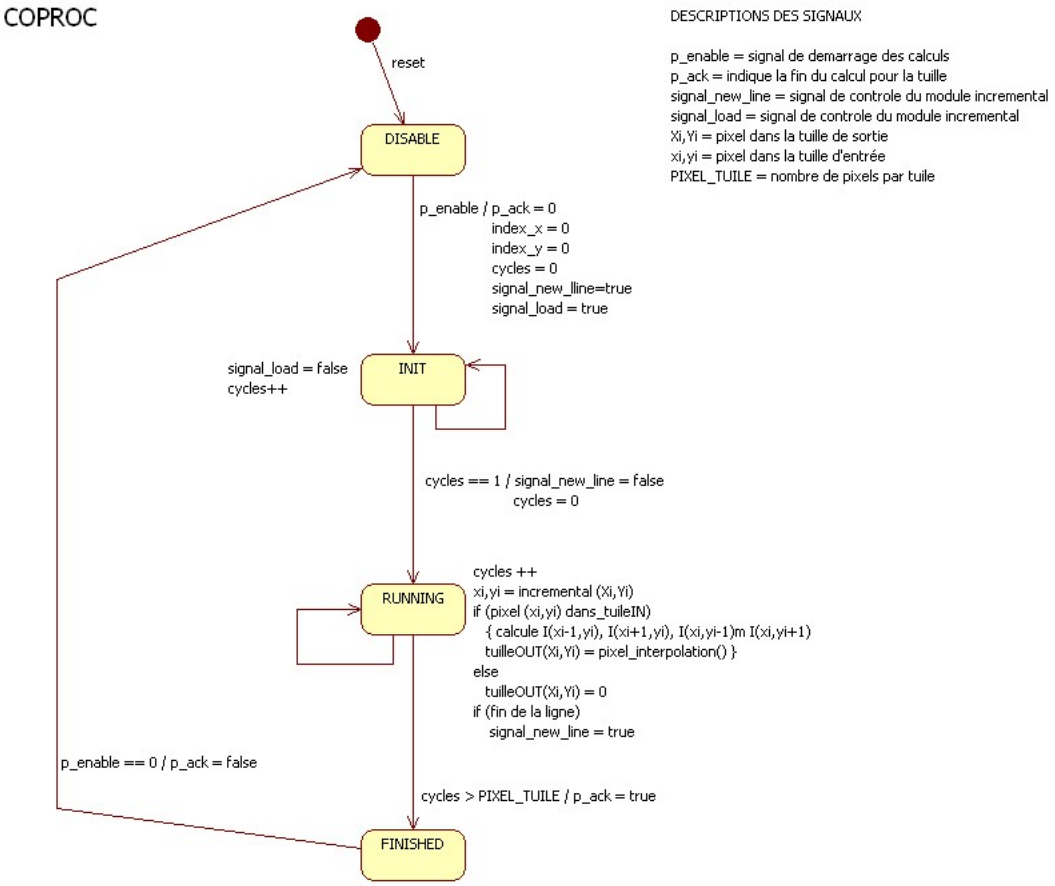


2.2 Fonctionement

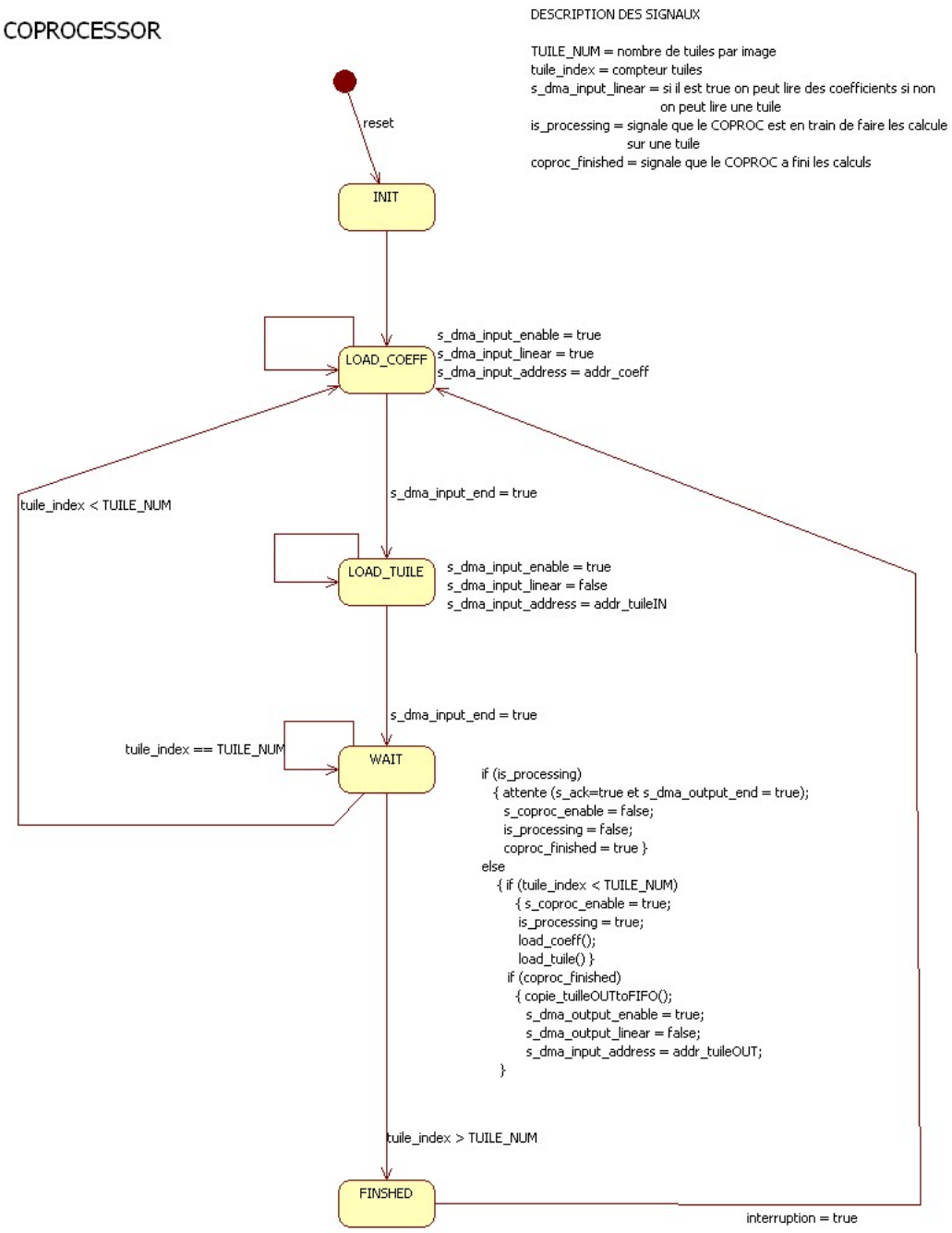
2.2.1 Sequence Diagram



2.2.2 Machine à état - COPROC



2.2.3 Machine à état - Coprocessor



3 Analyse

3.1 Chronogramme

La ligne en bleu (1) montre que l'interruption du Video_In démarre le Coprocessor. De l'autre côte, la ligne rose (2) montre la fin de la transformation, indiquée par le déclenchement de l'interruption du Coprocessor. La comparaison de ce temps avec celui d'envoi, représenté par FRAME_OUT_VAL, met en évidence la marge temporelle restante.

3.2 Résultats

	avec accès mémoire	sans accès mémoire	temps maximum à 60 fps
Transformation d'une image	14.03 ms	0.05 ms	16.67 ms

Le gros problème de ce système résides dans les accès mémoire car son débit est relativement faible ce qui impose une tuile d'entrée de faible taille. Malgré le taux d'utilisation mémoire inférieur à 100% (condition nécessaire), l'ordonnaçabilité n'est pas toujours garantie.

Le tableau au-dessus démontre que les accès sont vraiment contraignants, étant donné le temps total 280 fois plus long que celui de traitement pur.

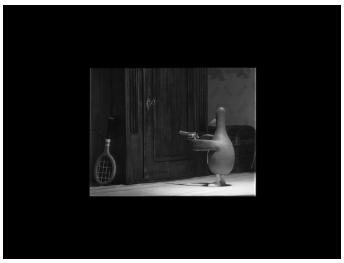
Si le temps est inférieur à 16.67 ms, le traitement respecte bien la contrainte de 60 fps.

3.3 Sortie de Transformations

Voici quelques exemples de transformations implémentées dans notre SoC :

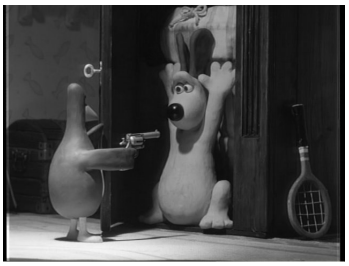
3.3.1 Zoom In

3.3.2 Zoom Out



3.3.3 Rotation

3.3.4 Miroir



4 Conclusion

Ce projet fut une une remarquable opportunité de pratiquer le travail en équipe et de développer des méthodes adéquates. Nous avons par exemple eu la chance de découvrir ou d'améliorer l'utilisation d'outils de développement partagés comme *mercurial*, la découverte d'éditeurs de texte professionnels tels que Vi ou de logiciels de composition de documents tes que TeX. Nous sortons donc grandis de cette formidable expérience avec une compréhension indéniablement accrue des systems on chips.

Références

[1] Renaud Pacalet, *Les formules de base du calcul incrémental*, 21 mai 2003.

[2] Tarik GRABA et Yves MATHIEU, liens : *Cours System on Chip - ENST*, juin 2010.