Optimisations structurelles et matérielles de l'encodeur vidéo H264/AVC sur un seul coeur d'un DSP multicoeurs TMS320C6472

Nejmeddine Bahri, Thierry Grandpierre, Nouri Masmoudi, Mohamed Akil

Optimisations structurelles et matérielles de l'encodeur vidéo H264/AVC sur un seul cœur d'un DSP multicoeurs TMS320C6472

NEJMEDDINE BAHRI[(1)], THIERRY GRANDPIERRE[(2)], NOURI MASMOUDI[(1)], MOHAMED AKIL[(2)]

[(1)] Ecole Nationale d'ingénieurs de Sfax, Université de SFAX, Tunisie
[(2)] ESIEE Engineering, Laboratoire d'informatique IGM, Université PARIS-EST, France

[1] nejmeddine.bahri@gmail.com, nouri.masmoudi@enis.rnu.tn [2] t.grandpierre@esiee.fr, akilm@esiee.fr

Résumé - Cet article présente une implémentation optimisée d'un encodeur vidéo H264/AVC sur un seul cœur d'un DSP à 6 cœurs TMS320C6472 pour des vidéos à basse résolution CIF (Common Intermediate format 352x288) dans le but de faire prochainement une implémentation multicoeurs SD (Standard Definition) et HD (High Definition). Vu la complexité de ce standard de compression vidéo, des optimisations structurelles et matérielles sont proposées afin d'accélérer la vitesse d'encodage dans le but d'atteindre le temps réel. L'exploitation de la grande taille de la mémoire sur puce afin de minimiser l'accès à la mémoire externe et l'utilisation de l'unité de transfert (EDMA) pour paralléliser le transfert de données avec le traitement permettent d'avoir un gain de 35% sur la vitesse d'encodage. Ces résultats d'implémentation optimisée de l'encodeur sur un seul cœur DSP à 700 MHz assurent l'encodage à 25 f/s pour la résolution CIF et valident la perspective d'atteindre le temps réel pour des résolutions plus élevées en passant à une implantation sur 6 cœurs.

Mots clés - H264/AVC, DSP TMS320C6472, optimisations structurelles et matérielles, EDMA, Temps réel.

1 Introduction

L'encodeur vidéo H264/AVC [1] est un standard de compression vidéo. Il assure un bon compromis entre l'efficacité de compression et la qualité vidéo. Cette efficacité est accompagnée par une grande complexité de calcul engendrant des problèmes pour les applications vidéo qui nécessitent un traitement en temps réel (25 à 30 f/s). Plusieurs solutions ont été proposées afin de réduire la complexité de calcul de cet encodeur. Ainsi, différents travaux ont essayé d'appliquer des optimisations algorithmiques afin de simplifier le traitement [2] [3] [4]. D'autres ont proposé d'accélérer certaines parties de l'encodeur à l'aide de parties dédiées matériellement [5] [6]. Enfin, certaines solutions ont exploité les nouvelles technologies multicoeurs que l'on peut maintenant trouver dans les processeurs embarqués afin de partitionner les tâches sur différentes unités de calcul pour paralléliser le traitement [7] [8]. La plupart de ces implantations engendrent une dégradation de la qualité visuelle et augmentent souvent le débit. En effet, pour gagner en temps d'exécution et en complexité ils n'implantent pas tous les modes de prédictions proposés par le standard (3 ou 4 modes sur les 9 à tester pour l'intra 4x4 par exemple), ils réduisent aussi la taille de la fenêtre de recherche pour l'estimation de mouvement. De plus les implantations parallèles découpent souvent l'image en slices traitées en parallèle sur chaque cœur mais ce découpage induit aussi une perte de qualité puisque les dépendances de traitement entre certains MBs ne sont plus respectées. Enfin, en désactivant l'option de « rate control » qui agit sur le facteur de quantification pour réduire la quantité d'information, une augmentation du débit ne sera plus négligeable. Dans ce cadre, nous avons choisi de commencer par une implantation CIF H264/AVC complète et optimisée sur un unique cœur d'un DSP multicoeurs. Ce travail, présenté dans ce papier, constituera le point de départ pour l'implémentation HD. En effet, pour passer à la HD, il nous faudra utiliser plusieurs cœurs pour atteindre le temps réel et obtenir un encodeur plus performant que les IPs existantes sur le marché tant au niveau de la qualité vidéo obtenue que du taux de compression (débit). Le reste de ce papier est structuré comme suit : après une brève introduction des principes de l'encodeur H264/AVC dans la section 2, l'architecture du DSP TMS320C6472 est présentée dans la section 3. Les optimisations structurelles et matérielles sont détaillées dans la section 4. Les résultats expérimentaux de l'implémentation sur un seul cœur DSP sont discutés dans la section 5 avant de conclure.

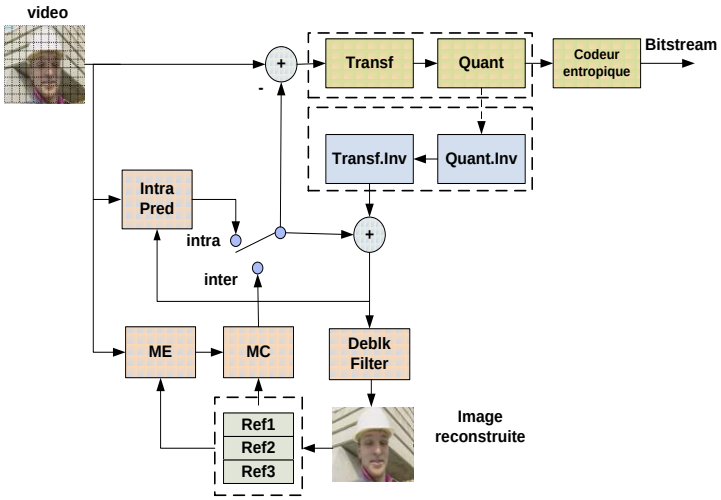

Figure 1 : Structure de l'encodeur vidéo H264/AVC

2 Principe de l'encodeur H264/AVC Baseline profile

La chaîne de codage vidéo H.264/AVC baseline profile, illustrée par la figure 1, est un hybride de prédictions temporelle et spatiale. L'image est d'abord

divisée en macroblocs (MB). Chaque MB va subir 2 types de prédiction :

❖ Une intra prédiction qui permet d'éliminer les redondances spatiales au sein de l'image. Elle se répartit en deux modes : intra16x16 et intra 4x4. Cette prédiction nécessite l'utilisation des pixels voisins en haut et à gauche de chaque MB traité.

❖ Une inter prédiction qui sert à réduire les redondances temporelles dans la vidéo. Elle consiste à déterminer le vecteur de mouvement d'un MB dans une image i par rapport à sa position dans de multiples images de référence i-n avec n est compris entre 1 et 16. La recherche de vecteurs de mouvements est limitée dans une zone qu'on appelle « fenêtre de recherche ». Cette fenêtre est constituée des MBs voisins qui entourent le MB courant mais dans l'image i-n déjà encodée à l'instant t-n.

Une transformée entière et une quantification sont appliquées sur le MB résiduel qui est la différence entre le MB original et le meilleur MB prédit parmi les 2 types de prédiction. Les coefficients transformés et quantifiés seront soumis à un codage entropique pour reconstruire le bitstream qui sera par la suite transmis (par radio, fibre etc) vers un décodeur ou bien stocké dans un fichier. La chaîne de décodage dans l'encodeur est utilisée pour faire la prédiction des blocs suivants. L'image reconstruite est transmise au filtre de déblocage (filtre anti-bloc) pour éliminer les artefacts. Ce filtrage nécessite l'utilisation des pixels voisins en haut et à gauche du MB reconstruit dans l'image reconstruite.

3 L'architecture du DSP TMS320C6472

Le TMS320C6472 fait partie des dernières générations de DSP multicoeurs fabriqués par Texas Instrument (TI). Très performant, il est caractérisé par une faible consommation électrique 0,15 mW / MIPS à 3 GHz qui le rend approprié pour de nombreuses applications embarquées. Six cœurs DSP C64x +, 4.8 Moctets (Mo) de mémoire sur puce, un jeu d'instructions "very long instruction word" (VLIW) et une fréquence de 700 MHz pour chaque cœur se combinent pour offrir une performance de 33600 MIPS. Chaque C64x + intègre une grande quantité de mémoire sur puce partagée à deux niveaux pour chaque cœur : mémoires niveau1 L1P et L1D de taille 32 Koctets (Ko) et mémoire locale niveau 2 (L2) partagée entre le programme et les données de taille 608Ko. La mémoire L2 peut également être configurée en tant que SRAM, cache, ou une combinaison de deux. Les 6 cœurs se partagent aussi une mémoire partagée L2RAM de 768 Ko ce qui permet d'éliminer l'accès à la mémoire externe DDR2, ainsi réduire la dissipation d'énergie et accélérer l'exécution puisque les mémoires sur puce sont plus rapides que les mémoires externes.

4 Optimisations structurelles et matérielles

Notre but est de commencer par élaborer un code H264/AVC bien optimisé sur un seul cœur DSP pour une résolution CIF avant de pouvoir passer à une implémentation SD et HD sur multicoeurs et exploiter ainsi le parallélisme potentiel présent dans la norme H264. Différentes optimisations mono cœur ont donc été proposées exploitant l'architecture interne du DSP TMS320C6472.

4.1 Optimisations structurelles

Elles consistent à concevoir une architecture qui exploite efficacement un cœur du DSP et surtout l'utilisation de la mémoire interne caractérisée par sa rapidité par rapport à la mémoire externe SDRAM. Chaque cœur possède une mémoire interne LL2RAM de taille 608 Ko partagée entre le programme et les données. De préférence et dans la mesure du possible nous devons donc y charger le programme et les données. Deux variantes sont proposées.

4.1.1 Architecture « MB par MB »

Cette architecture présente la structure standard du traitement des données dans la norme H264 basée sur l'encodage d'un MB suivi d'un autre jusqu'à terminer tous les MBs d'une image. Le principe est le suivant : le code H264 dont la taille est 120Ko est chargé dans la mémoire interne LL2RAM. Ainsi 488 Ko d'espace mémoire LL2RAM restent libres parmi les 608 Ko. Pour un format d'image YUV 4:2:0 utilisé dans le « baseline profile » (pour 4 pixels luminance Y, on a 1 pixel chrominance U et 1 pixel chrominance V) et pour une résolution CIF, l'image source (352 x 288 x 1.5=148.5 ko), l'image de référence étendue par un MB sur les 4 faces nécessaire pour l'estimation de mouvement ((352+32) x (288+32) x 1.5=180 ko), l'image reconstruite (180 ko), et le bitstream (64 Ko) ont été stockés dans la mémoire externe du DSP en raison de leur grande taille. Dans notre implémentation, on a choisi d'utiliser une seule image de référence afin de simplifier le calcul. Pour éviter de travailler directement sur la mémoire externe, les données nécessaires pour encoder un MB sont copiées de la mémoire DDR2 vers des buffers créés en mémoire interne. Parmi elles, on trouve, le MB source, la fenêtre de recherche et le MB reconstruit pour les 3 composantes YUV. Les autres données nécessaires pour l'encodage telles que les matrices de quantification et de transformée, les MBs prédits, les matrices de SAD sont aussi alloués dans la mémoire interne afin d'accélérer le traitement et de minimiser l'accès à la mémoire externe. La quantité totale des données allouées dans la mémoire interne est égale à 28.24 Ko. Donc 459.76 Ko sont encore libres. Le principe d'encodage d'un MB luminance Y pour cette architecture est le suivant (même principe pour la partie chrominance) : le CPU charge un MB source (16x16) et la fenêtre de recherche (9 MBs pour chaque MB source) de la mémoire DDR2, où il y a l'image source à traiter à cet instant t et l'image de référence qui est l'image traitée à l'instant t-1, vers la mémoire interne. Tout le traitement de la chaine H264 est effectué maintenant dans la mémoire interne du cœur DSP. Le MB reconstruit (20x20), étendu par 4 pixels en haut et 4 pixels à gauche nécessaires pour le filtrage,

sera transféré de la mémoire interne vers la mémoire externe dans l'image reconstruite. Le travail se répète jusqu'à terminer tous les MBs de l'image source.

L'avantage de cette architecture est qu'elle est presque indépendante de la résolution vidéo. Elle est adaptée à n'importe quel type de DSP même s'il n'a pas une mémoire interne importante (55.54 Ko sont nécessaires pour la résolution HD 720p (1280x720)).

Les inconvénients majeurs de cette architecture sont d'une part, l'accès important à la mémoire externe à chaque lecture d'un MB source, lecture de la fenêtre de recherche et à chaque écriture d'un MB reconstruit dans l'image reconstruite et aussi, la nécessité de sauvegarder, après chaque traitement d'un MB i, les voisinages gauche et haut pour faire la prédiction et le filtrage du MB i+1 d'une autre part.

4.1.2 Architecture « 1 ligne de MBs »

Une deuxième architecture a été conçue afin de réduire les inconvénients de la première architecture « MB par MB » ainsi, diminuer l'accès à la mémoire externe et éviter à chaque fois la sauvegarde des voisinages. Le principe de cette architecture est illustré par la figure 2.

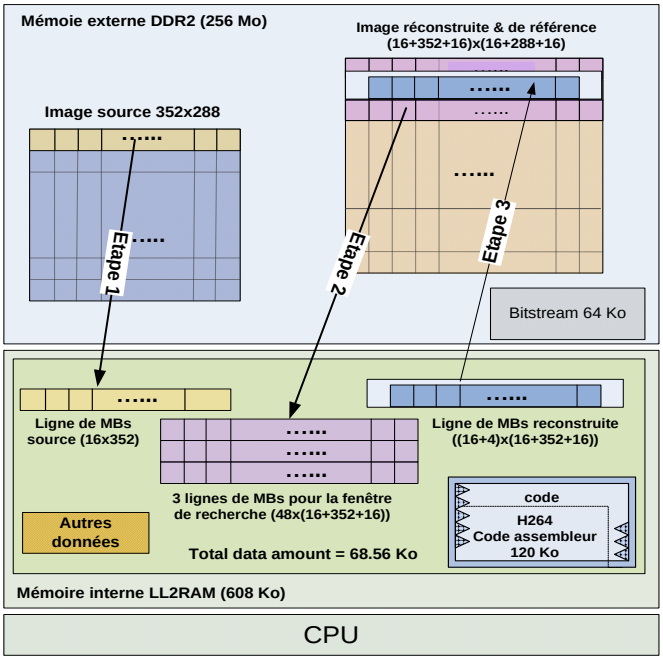

Figure 2 : Architecture « 1 ligne de MBs »

Elle consiste à faire la lecture d'une ligne de MBs source (16 x largeur_image) et 3 lignes pour la fenêtre de recherche (48 x (16+largeur_image+16)) de la mémoire externe vers la mémoire interne. Le CPU encode toute la ligne de MBs source sans accès à la mémoire externe et quand il termine le traitement, il transfère la ligne de MBs reconstruite (20 x (16+largeur_image+16)) de la mémoire interne du DSP vers la mémoire externe dans l'image reconstruite. Cette image a joué le rôle aussi de l'image de référence puisque les données écrasées ne sont plus utiles (elles sont déjà copiées dans les 3 lignes de la fenêtre de recherche). En passant à la deuxième ligne source, il n'est pas nécessaire de charger 3 lignes pour la fenêtre de recherche de l'image de référence, il suffit de décaler en haut les deux dernières lignes de la fenêtre de recherche dans la mémoire interne et amener la troisième ligne de la mémoire externe à partir de la quatrième ligne de l'image de référence et ainsi de suite. La quantité totale des données allouées dans la mémoire interne du cœur DSP pour une résolution CIF est 68.56 Ko au lieu de 28 Ko pour la première architecture. Cette architecture satisfait les contraintes mémoires, mais il faudra faire attention en augmentant la résolution pour travailler avec la qualité HD. Cette architecture a permis de réduire l'accès à la mémoire externe car, en considérant la résolution CIF (352x288) dont une ligne fait 22 MBs, on ne fait plus qu'un seul accès externe alors qu'il fallait 22 accès pour l'architecture « MB par MB ». De plus, on a éliminé la sauvegarde des voisinages gauche pour la prédiction et le filtrage d'un MB à l'abscisse X puisque ils sont déjà existants dans la ligne reconstruite à l'abscisse X-1. De même, on a réduit la sauvegarde des voisinages haut, contrairement à l'architecture « MB par MB » qui nécessite de faire cela pour chaque MB traité, cette architecture nécessite de faire la sauvegarde une seule fois après avoir terminé l'encodage de toute la ligne de MBs.

4.2 Optimisations matérielles

4.2.1 Architecture « 2 lignes de MBs » : utilisation d'EDMA

Le but de cette optimisation est de minimiser le temps de transfert des données de la mémoire externe vers la mémoire interne en utilisant le contrôleur EDMA du DSP (Enhanced Direct Memory Access) qui permet de transférer des données entre deux espaces mémoires différents. Le C6472 dispose d'un contrôleur EDMA3 possédant 64 canaux DMA et 4 canaux QDMA (Quick DMA). Ces canaux peuvent adresser 256 registres «PaRAM Sets» qui contiennent les informations de configuration de transfert telles que l'adresse source, l'adresse de destination, la taille des données à transférer, le mode de synchronisation...etc. Ces canaux peuvent être déclenchés par plusieurs méthodes (Event Triggering, manual Triggering et chain Triggering).

Cette optimisation est incluse dans la troisième architecture « 2 lignes de MBs » basée sur la notion classique des buffers ping pong dont le but est de paralléliser le transfert des données avec le traitement. Pour cela deux buffers ping pong ont été créés dans la mémoire interne : un pour les MBs sources et un pour les MBs reconstruits. Le principe de fonctionnement de cette architecture est décomposé en trois phases :

- La première phase : pendant que le CPU encode la première ligne de MBs source « ping », l'EDMA va amener la deuxième ligne « pong » en la transférant de la mémoire externe vers interne.
- La deuxième phase : le CPU fait le filtrage de la ligne reconstruite « ping », en parallèle, l'EDMA prépare les 3 lignes de la fenêtre de recherche pour la ligne de MBs source suivante puisque à la phase du filtrage, on n'aura pas besoin d'utiliser la fenêtre de recherche et de cette façon, ce n'est pas nécessaire d'utiliser un buffer ping pong pour la fenêtre de recherche.

Tab 1 : Evaluation des performances des architectures proposées

Vitesse d'encodage f/s / Séquence	Architecture « MB par MB »	Architecture « 1 Ligne de MBs »	Architecture « 2 Lignes de MBs »	Architecture «2 Lignes de MBs avec cache 256 Ko»
Foreman	16.28	22.03	23.70	24.83
Akiyo	16.78	23.06	24.94	25.87
News	16.94	23.37	25.27	26.37
Container	16.49	22.35	24.25	25.47
Vitesse moyenne (f/s)	**16.62**	**22.70**	**24.54**	**25.63**

❖ La troisième phase : l'EDMA fait le transfert de la ligne reconstruite « ping » de la mémoire interne vers externe et le transfert de la ligne source suivante « ping » de la mémoire externe vers interne en utilisant des canaux DMA différents, en parallèle, le CPU fait l'encodage de la ligne source « pong » etc.
Cette architecture consomme 88 Ko de données allouées dans la mémoire interne LL2RAM. Ainsi 400 Ko restent encore libres dans cette mémoire.

4.2.2 Activation de la mémoire cache

La mémoire locale du chaque cœur DSP TMS320C6472 peut également être configurée en tant que L2 SRAM, L2 cache, ou une combinaison de deux. Etant donné que l'on a encore 400 ko d'espace libre dans cette mémoire, on peut configurer une partie de cette mémoire comme cache afin d'accélérer le traitement et diminuer le temps d'accès (en lecture ou en écriture) du CPU à ces données. Pour le DSP TMS320C6472, la mémoire cache peut être configurée selon 4 voies : 32 Ko, 64Ko, 128Ko et 256 Ko. Dans ce cas, on peut choisir la valeur maximale du cache 256 Ko afin de minimiser la probabilité de « cache misses » c'est-à-dire l'absence des données dans la mémoire cache.

5 Résultats expérimentaux

Les quatre architectures présentées auparavant sont implémentées sur un seul cœur DSP à 700MHz en utilisant différentes séquences vidéo de test pour une résolution CIF (352x288). Le tableau 1 indique les vitesses d'encodage de chacune d'elles. L'architecture « MB par MB » est la plus lente, elle assure une vitesse d'encodage de 16.62 f/s qui est très loin du temps réel (25 f/s) à cause de l'accès trop long à la mémoire externe. La deuxième architecture que nous avons conçu pour surmonter les désavantages de l'architecture « MB par MB » a permis d'avoir un gain d'environ 26.78% par rapport à la première architecture proposée en arrivant à 22.7 f/s. L'utilisation d'EDMA dans la troisième architecture proposée « 2 Lignes des MBs » afin de paralléliser le transfert des données avec le traitement assure un gain de 7.5% par rapport à l'architecture « 1 ligne des MBs » et assurant une vitesse d'encodage de 24.54 f/s. L'activation de la mémoire cache avec l'utilisation d'EDMA nous a permis d'atteindre le temps réel 25 f/s et assurant un gain total par rapport à l'architecture « MB par MB » d'environ 35.15 %. En partant de cette implantation il est maintenant possible d'envisager le temps réel pour des résolutions plus élevées si l'on exploite le parallélisme potentiel de la norme H264 pour en faire une implémentation multicœurs. Il faudra donc explorer différentes méthodes de partitionnement (« GOP Level parallelism », « Frame Level parallelism »…), pour rechercher la plus adaptée.

De plus les caractéristiques texturales et morphologiques propres aux vidéos à haute définition permettent aussi d'envisager de nouvelles optimisations algorithmiques sur les modules les plus complexes que sont l'intra prédiction et l'inter prédiction.

6 Conclusion

Dans cet article, une implémentation optimisée de l'encodeur H264/AVC sur un seul cœur d'un DSP multicoeurs TMS320C6472 a été obtenue. Des optimisations structurelles et matérielles ont été proposées pour arriver à un codage en temps réel 25 f/s pour la résolution CIF (352x288). Nos optimisations permettent de réduire la durée d'encodage de plus de 35% et nous ont permis d'atteindre une vitesse d'encodage de 25,63 f/s.

Pour encoder des résolutions plus élevées, de type SD et HD 720p nous passerons à une implémentation sur les 6 cœurs du DSP en exploitant le parallélisme potentiel existant dans la norme H264/AVC. Nous profiterons des caractéristiques texturales et morphologiques des vidéos HD pour ajouter des optimisations algorithmiques.

Enfin, l'utilisation du dernier DSP multicoeurs de Texas Instrument, le DSP TMS320C6678 intégrant huit C66x cœurs, fonctionnant chacun à 1,25 GHz (presque double du C6472) devrait permettre d'atteindre l'encodage temps réel pour la résolution HD 1080p.

7 Remerciement

Ce travail est effectué dans le cadre d'une thèse en cotutelle entre l'École Nationale d'Ingénieurs de Sfax ENIS en Tunisie et ESIEE Paris en France. Il est soutenu par les ministères français des Affaires étrangères (MAE), de l'Enseignement supérieur et de la Recherche (MESR) et le Ministère tunisien de l'Enseignement Supérieur et de la Recherche

Scientifique (MESRS) dans le cadre du Partenariat Hubert Curien (PHC UTIQUE) sous le numéro du projet CMCU 12G1108.

8 Bibliographies

[1] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, "Draft ITU-T Recommendation and Final Draft international Standard of Joint Video Specification (ITU-T Rec. H.264 ISO/IEC 14496-10 AVC)", JVT-G050, 2003.

[2] Nejmeddine Bahri, Imen Werda, Amine Samet, Mohamed Ali Ben Ayed and Nouri Masmoudi, “Fast Intra Mode Decision Algorithm for H264/AVC HD Baseline Profile Encoder,” International Journal of Computer Applications volume 37-No.6, January 2012

[3] Kyungmin Lim, Seongwan Kim, Jaeho Lee, Daehyun Pak; Sangyoun Lee, "Fast block size and mode decision algorithm for intra prediction in H.264/AVC," Consumer Electronics, IEEE Transactions, vol.58, no.2, May 2012

[4] Dongil Han, Kulkarni Amruta, Rao, K.R, "Fast inter-prediction mode decision algorithm for H.264 video encoder," Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2012 9th International Conference pp.1-4, 16-18 May 2012

[5] Colenbrander, R.R.; Damstra, A.S.; Korevaar, C.W.; Verhaar, C.A.; Molderink, A.; , "Co-design and Implementation of the H.264/AVC Motion Estimation Algorithm Using Co-simulation," Digital System Design Architectures, Methods and Tools, 2008. DSD '08. 11th EUROMICRO Conference, p.210-215, 3-5 Sept. 2008

[6] Chih-Hung Kuo, Li-Chuan Chang, Kuan-Wei Fan, Bin-Da Liu, "Hardware/Software Codesign of a Low-Cost Rate Control Scheme for H.264/AVC," IEEE Transactions on Circuits and Systems for Video Technology Volume 20 Issue 2, February 2010

[7] M. Bariani, P. Lambruschini, M. Raggio, “An Efficient Multi-Core SIMD Implementation for H.264/AVC Encoder,” VLSI Design, vol. 2012, Article ID 413747, 14 pages, 2012.

[8] S.Sankaraiah, H.S.Lam, C.Eswaran and Junaidi Abdullah, "GOP Level Parallelism on H.264 Video Encoder for Multicore Architecture," 2011 International Conference on Circuits, System and Simulation IPCSIT vol.7 (2011) IACSIT Press, Singapore