# MUX-MCM Based Quantization VLSI Architecture for H.264/AVC High Profile Encoder

Jiang Ying, Xinhua Chen
College of Information Science and Engineering
Shandong University of Science and Technology
Qingdao, China

Yibo Fan, Xiaoyang Zeng
State Key Lab of ASIC and System
Fudan University
Shanghai, China
{fanyibo, xyzeng}@fudan.edu.cn

*Abstract*—**This paper presents a hardware-efficient and high-throughput quantization implementation for H.264/AVC high profiles encoder. The constant multiplication in quantization is accomplished by time-multiplexed multiple-constant multipliers (MUX-MCM). By rational pipeline decision, the proposed design manages to achieve a high throughput at a low area cost. Synthesized with SMIC0.18µm technology, the proposed design reaches a maximum operating frequency of 250Mhz with a throughput of 1Gpixels/sec at the hardware cost of 28.56 Kgates.**

***Keywords-quantization;multipl constant multiplier;MUX-MXM; H.264/AVC encoder***

## I. Introduction

The H.264/AVC standard published in May, 2003, was primarily focused on "entertainment-quality" video, based on 8-bits/sample, and 4:2:0 chroma sampling [1]. In order to meet the requirement of high definition (HD) application, the Fidelity Range Extensions was added to this standard in July, 2004. The FRExt amendment puts forward High Profiles including a set of four new profiles: The High profile (HP), The High 10 profile (Hi10P), The High 4:2:2profile (H422P) and The High 4:4:4 profile (H444P) ,which support all tools of the prior Main profile. One of the improvements is introducing the perceptual quantization scaling matrices [1] which is also called weighted quantization.

A uniform quantize step is used in the default H.264/AVC quantization process to all the coefficients in a sub-block [2]. However, it has shown that the sensitivity of the human visual system (HVS) to the distortions due to quantization varies with coefficient frequency. Frequency-dependent quantization can change the quantize step for every coefficient within a block by a customized scaling factor. This is only available in the High profiles of H264/AVC. JVT has proposed a default set of quantization scaling matrices in the reference software JM to initialize the quantization step which are compatible with the prior quantization and scale process.

Quantization is the step that causes signal loss in lossy compression. The residual coefficients after integer DCT are quantized by multiplying by a set of constants which are called Multiplication Factors in H.264/AVC. A periodic quantization step table is used so as to define a large range of Quantization Parameters (QP) without increasing the memory requirements. For 9-bit prediction residuals, all the operations can be complemented by 16-bit arithmetic except the multiply operation. However, multiplication is a high cost operation in digital signal process system.

To reduce the area and power consumption in quantization process, [4][5] proposed a low-complexity quantization scheme by modifying the quantization parameter into smaller bit-width adders , which leads to mismatch error between encoder and decoder inevitably. Reference [6] proposed a 4-stage pipeline multiplier to increase the speed for quantization of 8x8blocks. A compression-guided quantization with a thresholds compare module was proposed in [7].The compare module predicts the coefficients that can be set as 0 directly in a 4x4 block to cut down the multiply operations. In [8], the area optimized architecture processes coefficients one by one with only one quantization unit, while the speed optimized architecture processes all the 16 coefficients in a 4x4block in parallel. 4x4block and 8x8block quantization are accomplished in [9] with a high parallelism and additional registers. Reference [10] explores the tradeoff between area cost and performance in terms of different degree of parallelism. Except [4][5], in all the other schemes mentioned above, the multiplications are complemented by general multipliers and lookup table.

The multiply operation in quantization is a type of multiple constants multiplication (MCM) problem [11].However, time-multiplexed multiple-constant multiplication (MUX-MCM) [12][13] is more suitable. This paper presents a high efficient quantization architecture based on MUX-MCMs for 4x4 coefficients blocks after integer DCT in H.264/AVC high profiles encoder.

The rest parts of this paper are organized as follow. The main aspects of quantization and scale in High Profiles of H.264/AVC Standard are presented in section II. Section III describes the details of the proposed design. The synthesis result and comparison with the previous works are presented in section IV and Section V concludes the paper.

## II. 4X4QUANTIZATION AND SCALING IN H.264/AVC HIGH PROFILE

H.264 assumes a scalar quantization [14]. The basic forward quantization formula is:

$$Z_{ij} = round(Y_{ij} / Qstep) \qquad (1).$$

978-1-4577-0170-2/11/$26.00 ©2011 IEEE

Where $Y_{ij}$ is a coefficient after integer DCT, Qstep is a quantize step and $Z_{ij}$ is a quantized coefficient. 52 values of Qstep indexed by QP are supported by the standard.

TABLE I. QUANTIZATION STEP SIZES IN H.254/AVC CODEC[2]

| **QP** | **0** | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|---|
| **QStep** | 0.625 | 0.6875 | 0.8125 | 0.875 | 1 | 1.125 |
| **QP** | **6** | **7** | **8** | **9** | **10** | **11** |
| **QStep** | 1.25 | 1.375 | 1.625 | 1.75 | 2 | 2.25 |
| **QP** | **…** | **…** | **…** | **…** | **…** | **…** |
| **QStep** | … | … | … | … | … | … |
| **QP** | **48** | **49** | **50** | **51** | | |
| **QStep** | 160 | 181.333 | 202.666 | 224 | | |

Table I shows the QP and the corresponding Qstep. Qstep doubles for every increment of 6 in QP.

In order to keep the orthogonality of the integer DCT, a post-scale matrix PF is incorporated to the quantization step. Eq. (1) is written as:

$$Z_{ij} = round(W_{ij} * PF_{ij} / Qstep) \qquad (2).$$

To avoid division operations, (2) can be equivalently written as :

$$Z_{ij} = round(W_{ij} * MF_{m,S_{ij}} / 2^{qbits}),$$
$$S_{ij} = \begin{bmatrix} 0 & 2 & 0 & 2 \\ 2 & 1 & 2 & 1 \\ 0 & 2 & 0 & 2 \\ 2 & 1 & 2 & 1 \end{bmatrix}, m = QP\%6, i, j = 0...3 \qquad (3).$$

Quantization factors depend on the coefficients position (i,j) in a 4x4block. Matrix MF is shown in (4):

$$MF = \begin{pmatrix} 13107 & 5243 & 8066 \\ 11916 & 4660 & 7490 \\ 10082 & 4194 & 6554 \\ 9362 & 3647 & 5825 \\ 8192 & 3355 & 5243 \\ 7282 & 2893 & 4559 \end{pmatrix} \qquad (4).$$

For weighted quantization, a customized matrix called Weight_Scale is introduced to weight the quantization steps so that coefficients at different frequency may have different quantize step. The final quantization and scale factor matrix LevelScale is defined as:

$$\mathrm{LevelScale}_{m,\mathrm{i,\,j}} = \mathrm{MF}_{m,S_{\mathrm{ij}}} * 16 / \mathrm{Weight\_Scale}_{\mathrm{ij}} \qquad (5).$$

Eq.(3) is further modified as:

$$Z_{ij} = round(W_{ij} * LevelScale_{m,i,j} / 2^{qbits}) \qquad (6).$$

When Weight_Scale$_{i,j}$ = 16 (i=0…3,j=0…3), it is the original quantization mode which is called Flat Mode.

In integer arithmetic, (6) can be implemented as follows:

$$\begin{aligned} |Z_{ij}| &= \left(|W_{ij}| * \mathrm{LevelScale}_{m,i,j} + f\right) >> \mathrm{qbits} \\ \mathrm{Sign}(Z_{ij}) &= \mathrm{Sign}(W_{ij}) \end{aligned} \qquad (7).$$

f is used to compensate the visual effect of the video and is chosen by encoders. The values of f and qbits are different with the coefficients state [17]:

1) 4x4 Intra luma and chroma :

qbits = 15 + qp/6,f = 1<<(14+qp/6).

2)4x4 Inter:

qbits = 15 + qp/6,f = 1<<(13+qp/6).

3) 4x4 luma DC and chroma DC:

qbits = 16 + qp/6,f = 1<<(15+qp/6).

Specially, for DC coefficients including that of luma and chroma components, only LevelScale$_{m,0,0}$ is applied.

## III. PROPOSED QUANTIZATION USING MUX-MCMS

### A. Directed Acyclic Graph(DAG)

Reference [15][16]has proposed the minimized adder graph(MAG) to formulate constant integer multiplication. In MAGs, the shift and reuse of the results of intermediate additions is allowed. Fewer adders are required than the frequently-used Canonic Signed Digit (CSD) multipliers. For 19-bit constants, an improvement of 25% is available[16].

The multiplier implementation using MAG can be visually illustrated by directed acyclic graph (DAG). For demonstration, DAGs of the 6 constants of MF$_{m,1}$ are shown in Fig.1. The label on an edge is a positive 2-power integer, which represents the scaling (shifting toward left) applied to the operand on the edge. The label on a node represents the intermediate results of the addition or subtraction. Taking MF$_{0,1}$ as an example:

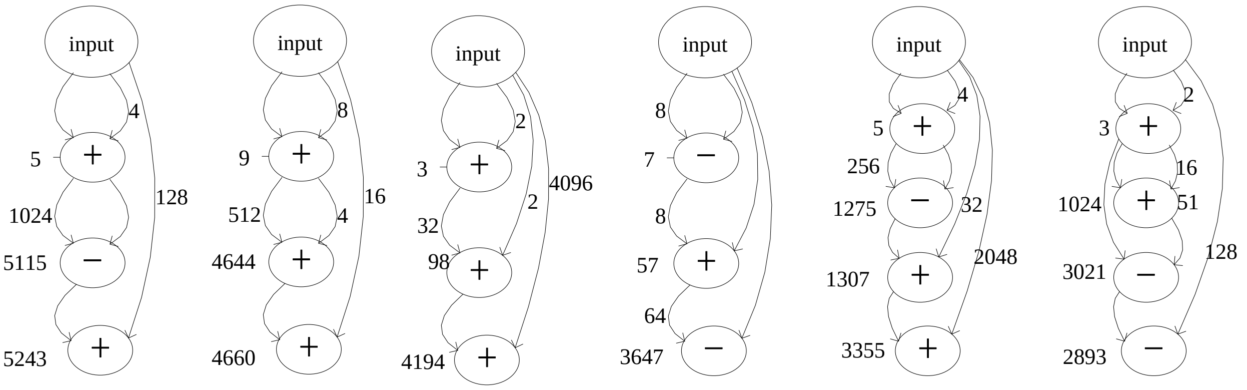
$$5243x = (1024(x + 4x) - (x + 4x)) + 128x \qquad (8).$$

### B. Multiplier Algorithm

Common subexpression elimination (CSE) is commonly applied to a set of constant multipliers that multiply a common variable [11]. It is more suitable for parallel multiplication.

But in the quantization and scaling process of H.264/AVC, coefficients after integer DCT only need to be multiplied by one of the six LevelScale controlled by QP%6 at one (i,j) position. And only one product is obtained at one time. The operation just like only one multiplier is presented. Three different methods of completing a multiple-constant multiplier are summarized in [13].The block diagrams of the three methods are depicted in Fig2.

In Fig. 2(1), a lookup table for constants and a general multiplier are used. QP%6 (the control signal) selects the constant to be fed into the input of the multiplier. In Fig. 2(2), CSE is used for the parallel multiplication and a multiplexer is used to select one of the parallel products according to the control signal. In Fig. 2(3), multiplexers are inserted to share the same adders at different time to reduced area requirements.



(1)$MF_{0,1}$ = 5243 (2)$MF_{1,1}$ = 4660 (3)$MF_{2,1}$ = 4194 (4)$MF_{3,1}$ = 3647 (5)$MF_{4,1}$ = 3355 (6)$MF_{5,1}$ = 2893

Figure 1 DAGs of MFm,$_1$

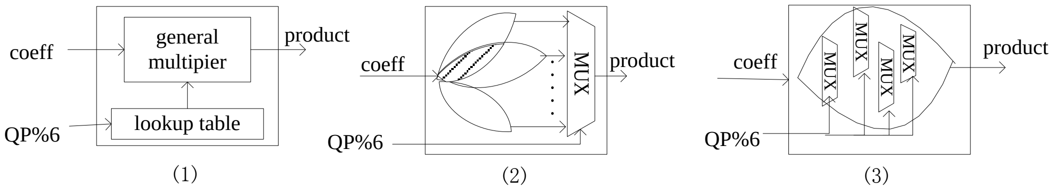


Figure 2.Three Implements of Multiple-Constant Multiplier[13]

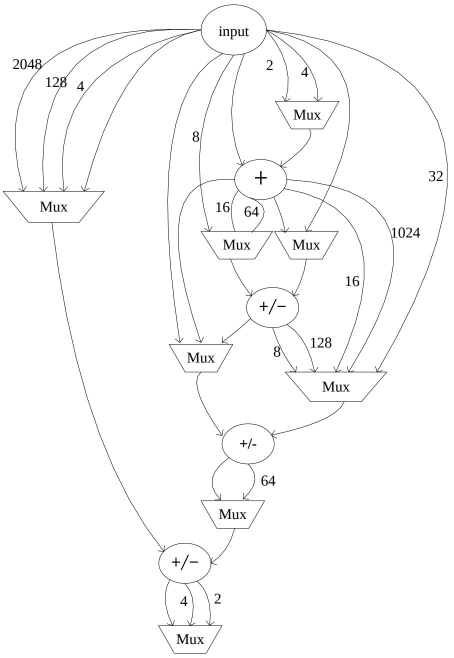


Figure 3 Merged DAGs of $MF_{m,1}$

Reference [13] has also proposed an algorithm to merge several separately DAGs into one to realize the time-multiplexed multiple-constant multiplication in Fig.2 (3). The number of adders required is as large as that of the largest initial DAG being merged. The merged DAG for the 6 separate DAGs of $MF_{m,1}$ is shown in Fig.3.

Three MUX-MCMs mapping to the three merged DAGs for $MF_{m,0}$, $MF_{m,1}$ and $MF_{m,2}$ are accomplished in this work.

### C. Proposed Quantization and Scale Architecture

The proposed quantization and scale architecture processes the results of 2D 4x4 integers DCT column by column and the quantized level are obtained according to (7). Four MUX-MCM instances are selected by the column index of a 4x4block simultaneously and four coefficients can be processed in parallel.

In this design, all of the elements in Weight_Scale are set as 16, so (7) can be reduction to (3).The diagram of the proposed quantization and scale module is shown in Fig.4. A four-stage pipeline including the control stage, multiplication stage and the post-process stage is adopted.

The control circuit produces the signals that are needed in the subsequent quantization arithmetic: the absolute values of the input coefficients, qbits, QP%6 and f defined in (7), as well as the column number indicator: column_r. The sign bit of $coeff_i$ (i=0,1,2,3) and column number are delayed two clock cycles to keep the synchronous with that in the multipliers. The MUX-MCM ARRAY module completes the main multiply operation in the quantization process according to the column number and the coefficients state.

To balance the critical path, the combinational logic of the multiplier deriving from the merged DAGs is divided into 2-stage pipeline. A correction factor f is added to the result of the multiplication depending on the coefficients state. Finally, the results are right shifted by qbits and restored the sign bit.

The bit width of the absolute value of input coefficients is 15. The largest multiply factor in MF is 13107 which is a 14-bit positive integer. The width of the products from MUXMCM ARRAY is 29 bits. When qbits is 15, the quantized levels have the biggest bitwidth 15. The results of quantization are 16 bits signed numbers.

TABLE II. SLECTION OF MUX-MCM

| Input Coefficients | DC | Others | |
|---|---|---|---|
| | ***Column0,..,3*** | ***Column0,2*** | ***Column1,3*** |
| Coeff0 | Mult0_0 | Mult0_0 | Mult2_0 |
| Coeff2 | Mult0_2 | Mult0_2 | Mult2_1 |
| Coeff1 | Mult0_1 | Mult2_0 | Mult1_0 |
| Coeff3 | Mult0_3 | Mult2_1 | Mult1_1 |

With the proposed design, every 4x1block is processed at the same time and has a latency of 4 cycles, and every 4x4block is processed in 7 cycles due to the 4-stage pipeline.

The details of the multiplier array in Fig.4 are illustrated in Fig5. Mult0_i, Mult1_i and Mult2_i are MUX-MCM instances for $MF_{m,0}$, $MF_{m,1}$, $MF_{m,2}$ respectively. Label i is the instance identifier of the multipliers. The multiplex controllers (MC) select the multipliers for the 4 input coefficients according to the column index and the coefficient state.

When the current coefficients are the 4x4DC coefficients block of a intra16x16 macro block, all of the coefficients are multiplied by $MF_{m,0}$ ,so four instances of $MF_{m,0}$ are need. In the case of all the other 4x4 coefficients block, two different multiply factors are used. When the column number is 0 or 2, two instances of both $MF_{m,0}$ and two $MF_{m,2}$ are needed. When the column number is 1 or 3, two instances of both $MF_{m,2}$ and two $MF_{m,1}$ are needed. Though $MF_{m,2}$ is required in each column with different coefficient position (i,j), two instances are enough. MC selects the proper MUC-MCM for every input coefficient .The selection result is shown in Table II.

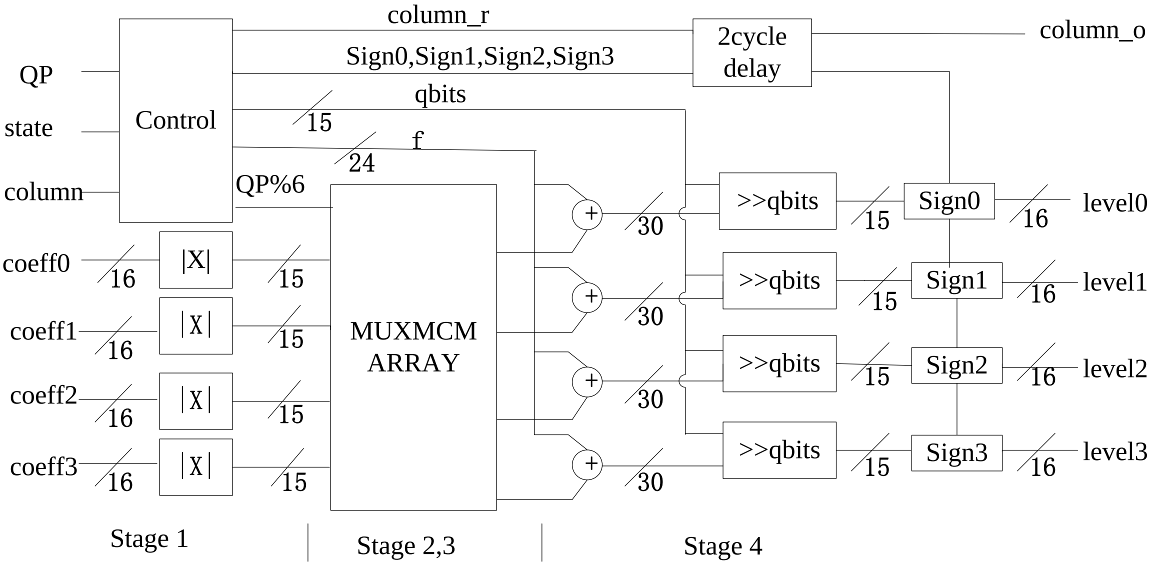


Figure 4. Details of MUX-MCM ARRAY

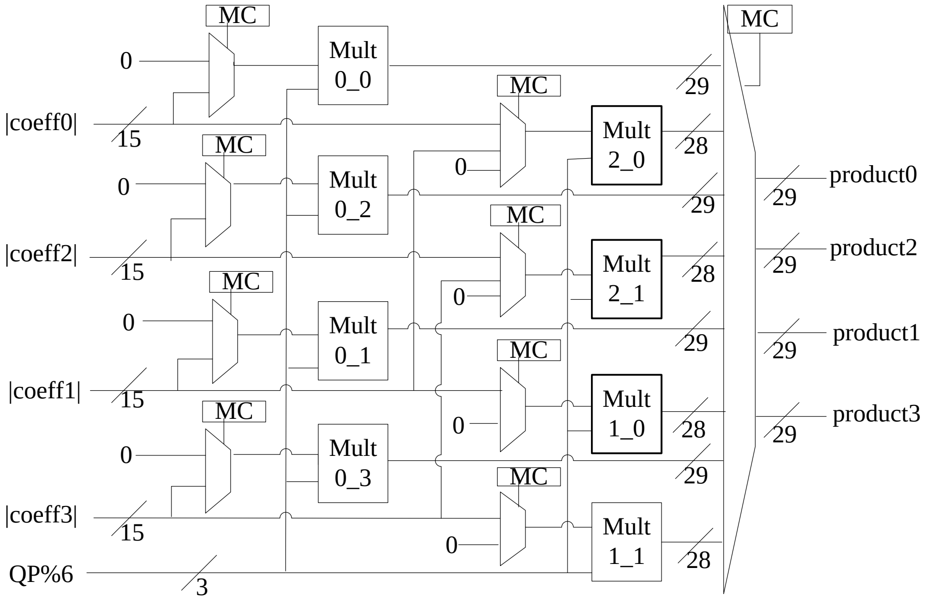


Figure 5. Diagram of the quantization and scale module

## IV. IMPLEMENTATION AND RESULTS

The proposed architecture is implemented in Verilog HDL, simulated in ModeSim-Altera-6.5e and synthesized by Synopsys Design Complier and SMIC0.18μ m technology.

Table III shows the performance of MUX-MCM compared with the multiple constants multipliers implemented by constants lookup table and general multiplier used in quantization procedure. All of these multipliers are accomplished with combinational logic. The MUX-MCM is about 50% smaller than general proposal with a slightly longer critical path. But the speed can be improved by pipeline technology with a few additional registers.

The comparisons with the published designs are shown in table IV. The proposed scheme can reaches a frequency of 250Mhz while all the prior designs completed in ASIC technology works at less than 100Mhz.The throughput of the speed optimized architecture in [8] is similar to the proposed design, but it works at only 68Mhz and the area is about 25% larger. Though the design proposed in [9] can achieve a higher throughput, the cost of area is 3 to 5 times higher with 32 coefficients processed at the same time.

TABLE III. COMPARISON ON DIFFERENT MULTIPILERS

| **Constant Group** | **General Multiplier+LUT** | | **MUX-MCM** | |
|---|---|---|---|---|
| | *Area(um$^2$)* | *Latency(ns)* | *Area(um$^2$)* | *Latency(ns)* |
| MFm,0 | 34624.56 | 4 | 11259.50 | 7.32 |
| MFm,1 | 29834.53 | 3.78 | 16439.49 | 6.95 |
| MFm,2 | 29990.86 | 3.98 | 16765.00 | 7.53 |

TABLE IV. COMPARISON WITH PRIOR WORKS

| **Solution** | **Technology** | **Area** *(Kgates)* | **Frequency** *(Mhz)* | **Multiplier number** | **Throughput** *(Mpixels/sec)* | **Throughput /Gates** |
|---|---|---|---|---|---|---|
| [8](area) | Tsmc0.35 | 1.749 | 85.47 | 1 | 20 | 11.44 |
| [8](speed) | Tsmc0.35 | 39.89 | 68 | 16 | 1058 | 26.5 |
| [9] | Tower0.18 | 104.05 | 76 | 32 | 2432 | 23.37 |
| | Ams0.35 | 77.668 | 79 | 32 | 2528 | 35.6 |
| [10] | Xilinx FPGA | 965 LUTs | 107 | 4 | 430 | - |
| Proposed | Smic0.18 | 28.561 | 250 | 8 | 4*250 | 35 |

## V. CONCLUSIONS

This paper presents a cost effective and high throughput quantization implementation for H.264/AVC high profiles encoder. By time-multiplexed multiple constants multipliers, this design achieve a high area advantage. By rational pipeline decision, the design manages to achieve a throughput of 1Gpixels/s at a frequency of 250Mhz.

## REFERENCES

[1] G. Sullivan, P. Topiwala, A. Luthra, “The H.264/AVC advanced video coding standard: overview and introduction to the fidelity range extensions”, in: SPIE Conference on Applications of Digital Image Processing XXVII, 2004.

[2] Iain Richardson,The H.264 Advanced Video Compression Standard(the second editon),John Wiley & Sons, 2010.

[3] H. S. Malvar, A. Hallapuro, M. Karaczewicz, and L. Kerofsky, “Low-Complexity Transform and Quantization in H.264/AVC”, IEEE Transactions on Circuits and Systems for video technology, vol. 13, no. 7, July 2003.

[4] Yun Zhang, Gangyi Jiang and Mei Yu, “Low-complexity quantization for H.264/AVC”, J Real-Time Image Proc (2009) 4:3–12.

[5] M.N. Michael, K.W. Hsu, “A Low-power Design of Quantization for H.264 Video Coding Standard”, in: Proceedings of the IEEE International SOC Conference, pp. 201–204, September 2008.

[6] Juan A. Michell,Jose M. Solana, Gustavo A. Ruiz, “A High-Throughput ASIC Processor for 8x8 Transformcoding in H.264/AVC”, Signal Processing: Image Communication Vol.26, Issue 2, pp. 93-104, February 2011.

[7] [7] J.D. Bruguera, R.R. Osorio, “A Unified Architecture for H.264 Multiple Block-size DCT with Fast and LowCost Quantization”, in: Proceedings of the 9th EUROMICRO Conference on Digital System Design, pp. 407–414,2006.

[8] [8] R.C. Kordasiewicz and S. Shirani, "ASIC and FPGA Implementations of H.264 DCT and Quantization Blocks",IEEE International Conference on Image Processing, 2005, pp. III - 1020-3, vol. 3, Genova, Italy, 11-14 September 2005.

[9] [9] G. Pastuszak, “Transforms and Quantization in the High-throughput H.264/AVC Encoder based on Advanced Mode Selection", in: Pro- ceedings of the IEEE Computer Society Annual Symposium on VLSI, pp. 203–208 ,April 2008.

[10] [10] Husemann R., Majolo M, et al., “Hardware Integrated Quantization Solution for Improvement of Computational H.264 Encoder Module”,in Pro- ceedings of the IEEE VLSI-SoC , ,pp. 316 – 321,27-29 September 2010.

[11] [11] K. Parhi, VLSI Digital Signal Processing Systems,Wiley, 1999.

[12] [12] P. Tummeltshammer, J. Hoe, and M. Püschel, “Multiple Constant Multiplication by Time-multiplexed Mapping of Addition Chains,” in Proc. Des. Autom. Conf., vol. 41, pp. 826–829,June 2004.

[13] [13] P. Tummeltshammer , J. Hoe and M. Pschel "Time-multiplexed Multiple-constant Multiplication", IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 26, no.9. pp. 1551-1563, 2007.

[14] [14] Iain Richardson,H.264 & MPEG-4 Video Compression, Wiley,2003.

[15] [15] A. G. Dempster and M. D. Macleod, “Constant integer Multiplication Using Minimum Adders,” IEE Proc. Circuits Devices Syst., vol. 141 no. 6, pp. 407–413, October 1994.

[16] [16] Gustafsson, A. Dempster, and L. Wanhammar, “Extended Results for Minimum-adder Constant Integer Multipliers,” in Proc. IEEE Int. Symp Circuits Syst., vol. 1, pp. I-73–I-76, May 2002.

[17] [17] A. Hallapuro, M. Karczewicz and H. S. Malvar, “Low Complexity Transform and Quantization – Part I: Basic Implementation,” Doc. JVT-B038, Geneva, CH, 29 January - 1 February 2002.