A Dynamically Reconfigurable DCT Architecture for Maximum Image Quality Subject to Dynamic Power and Bitrate Constraints

Yuebing Jiang and Marrios Pattichis
Department of Electrical and Computer Engineering
University of New Mexico
Albuquerque, New Mexico
{*yuebing, pattichis*}*@ece.unm.edu*

***Abstract*—We introduce a dynamically reconfigurable DCT architecture system that allows us to select optimal implementations based on bitrate and dynamic power constraints. For our approach, we first compute Pareto-optimal hardware realizations that are assessed in terms of reconstructed image quality, bitrate and dynamic power. The space of Pareto-optimal realizations are generated by varying both the number of non-zero DCT coefficients and the quality factor for the quantization table. From the generated hardware realizations, we then select the Pareto-optimal cases and discard all other cases. For each bitrate and dynamic power constraint, we use a dynamic partial reconfiguration (DPR) controller to implement the optimal DCT architecture.**

We test our approach using leave-one-out on the LIVE database, and implement our system on a Virtex-5 FPGA and demonstrate its performance using different bitrate and dynamic power constraints.

***Keywords*-FPGA, DCT, JPEG, dynamic partial reconfiguration.**

I. Introduction

Requirements for image compression can vary based on network conditions, available power and required levels of image quality. It is important to note that these requirements are time-varying. For example, network conditions can vary significantly, leading to the need to compress for different bitrates. Similarly, power consumption on mobile devices needs to be controlled to allow for longer operating periods. Also requirements on image quality can vary with the needs of the users. In this paper, we provide an optimization framework that allows for dynamic hardware reconfiguration to meet real-time constraints on bitrate and dynamic power while maximizing image quality. This is achieved by selecting Pareto-optimal realizations.

While the basic optimization framework can be applied to different image and video compression architectures, we focus on the DCT. Here, we map bitrate and dynamic power constraints to the number of zonally-encoded (zig-zag encoded) DCT coefficients and quality factors (QF). Then, the DCT hardware is dynamically reconfigured based on the required number of DCT coefficients. The architecture framework is implemented on an FPGA using dynamic partial reconfiguration (DPR). In this paper, we will assume that an efficient DPR controller is available. For an implementation of a high-speed DPR controller, we refer to [6].

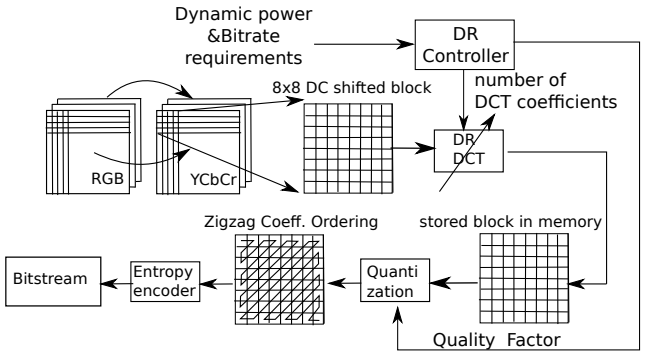

Figure 1. Dynamically reconfigurable DCT architecture for JPEG compression.

The basic idea of zonal coding applied to JPEG appeared in [3]. A hardware implementation of rectangular DCT region encoding appears in [4]. The authors in [4] also propose the use of Dynamic Partial Reconfiguration (DPR) for implementing the rectangular DCT-region encoding in digital videos. However, these prior approaches did not consider real-time constraints, as we do here.

The rest of the paper is organized as follows. In section II, we describe the proposed approach. The results are given in section III. Concluding remarks are then given in section IV.

II. Methodology

The proposed system is presented in Fig. 1. Real-time constraint in bitrate is input into the dynamic reconfiguration (DR) controller that is used to control the DCT hardware implementation and quality factor (QF). The DR DCT block is used to process images in 8×8 blocks that are later quantized based on the supplied QF. Each block is then zig-zag encoded using run-length encoding and standard Huffman tables.

Let (B_i, DP_i) for $i = 1, 2, \ldots, P$, denote a sequence of real-time constraints on the maximum bitrate and dynamic

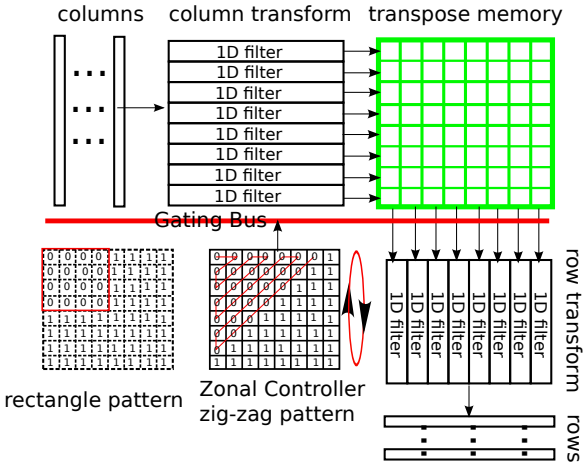

Figure 2. DCT Architecture Implementation.

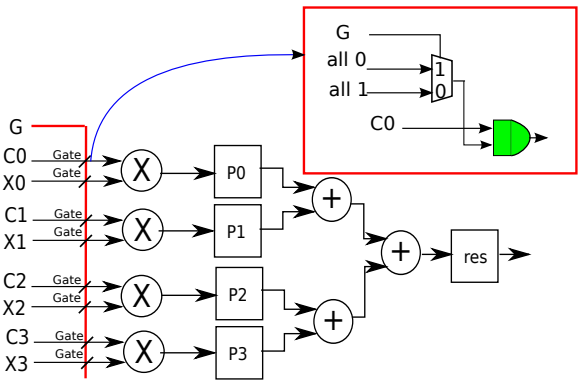

Figure 3. 1D DCT row filter using 4 multipliers and 3 adders.$C[0-3]$ and $X[0-3]$ are matrix coefficients and input data in eq(1) and eq(2). G is the gating signal from gating bus.

power respectively. To meet these constraints, we rely on doing the optimization off-line. This is possible for a small number of optimization problems (in the tens), where the optimal architectures can be computed and stored in memory for real-time reconfiguration. From these pre-computed solutions, we select the optimal DCT architecture that maximizes the SSIM [7] as given by:

$$\begin{aligned} \max_{\mathcal{A}(N_i),QF_i} \quad & \text{SSIM}\left(\mathcal{A}(N_i), QF_i\right) \\ \text{subject to:} \quad & \text{Bitrate}\left(\mathcal{A}(N_i), QF_i\right) \leq B_i, \\ & \text{Power}\left(\mathcal{A}(N_i), QF_i\right) \leq DP_i \end{aligned}$$

where N_i is the number of zonally-encoded DCT coefficients, QF_i denotes the Quality factor for the i-th constraint, and $\mathcal{A}(N_i)$ denotes the DCT architecture.

To solve this optimization problem, we will have to have prior estimates of the bitrate, dynamic power and SSIM as functions of the number of the (zonal) DCT coefficients and the quality factor. For estimating these functions, we use the 29 reference images from the LIVE database [2]. Then, for each pair (N, QF), where $N = 8, 16, \ldots, 64$, and $QF = 5, 10, \ldots, 95$, we take the median of bitrate and SSIM values over the training set database as robust estimates of the birtrate, dynamic power, and SSIM values. Since our prior estimates rely on the median values, it is quite possible that our estimated architectures will violate the bitrate constraints. To avoid real-time bitrate constraint violations, during off-line optimization, we use a more conservative constraint of $0.75 \cdot B_i$. As we shall describe in the results, this approach helps guarantee that over 75% of the architectures will meet the bitrate constraints without a need to dynamically reconfigure to another DCT architecture. The approach is validated using leave-one-out. That is, we use 28 images for the training set and the remaining image for testing.

Each DCT architecture is dynamically implemented using a DPR controller. The DPR controller is implemented on FPGAs using the internal configuration access port (ICAP). In this work, we used the standard Processor Peripheral partial reconfiguration process provided by Xilinx [8].

A. Fast DCT implementation

We consider a classic method for fast DCT implementation based on Chen's algorithm [1]. Chen's algorithm is based on a matrix factorization of the 1-D DCT transform. As we shall describe in the following section, the algorithm allows us to compute select DCT coefficients associated with zonal coding. Here, for completeness, we provide some of the details.

The 2D-DCT is implemented using $Z = (M \cdot X) \cdot M^T = Y \cdot M^T$, where X is the 8×8 input block, M is the forward DCT matrix, Y is contains the 1D DCT transforms along the columns (saved in transpose memory), and Z is the 2D output. Chen's algorithm factors the 8×8 matrix into two 4×4 matrices as given by:

$$\begin{bmatrix} Y(0,:) \\ Y(2,:) \\ Y(4,:) \\ Y(6,:) \end{bmatrix} = \begin{bmatrix} a & a & a & a \\ c & f & -f & -c \\ a & -a & -a & a \\ f & -c & c & -f \end{bmatrix} \begin{bmatrix} X(0,:)+X(7,:) \\ X(1,:)+X(6,:) \\ X(2,:)+X(5,:) \\ X(3,:)+X(4,:) \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} Y(1,:) \\ Y(3,:) \\ Y(5,:) \\ Y(7,:) \end{bmatrix} = \begin{bmatrix} b & d & e & g \\ d & -g & -b & -e \\ e & -b & g & d \\ g & -e & d & -b \end{bmatrix} \begin{bmatrix} X(0,:)-X(7,:) \\ X(1,:)-X(6,:) \\ X(2,:)-X(5,:) \\ X(3,:)-X(4,:) \end{bmatrix} \quad (2)$$

where $b = C_1, c = C_2, d = C_3, a = C_4, e = C_5, f = C_6, g = C_7,\ C_i = 0.5 \cdot cos(i\pi/16)$.

B. Zonal Coding

For zonal coding, note that the 2D DCT given by $Z = (M \cdot X) \cdot M^T$ is implemented by first applying a column-transform and then a row transform. For zonal coding, we

always compute the DCT column-transform. However, we selectively disable unneeded row-transforms based on the required number of 2D DCT frequencies. This second set of 2D DCT frequencies is selected based on their zig-zag order [3]. Alternatively, a rectangular ordering of the 2D DCT coefficients is given in [4]. Here, at about the same number of N_i, we note that zig-zag encoding can achieve average bitrate reduction of 8.9% and SSIM improvement of 0.72% over rectangle encoding for the entire LIVE database at $QF = 100$.

C. Hardware implementation

The proposed hardware implementation is shown in Fig. 2. Image columns are combined as shown in the RHS of (1) and (2). The 1D filters shown in Fig. 2 represent the rows in (1) and (2). The zonal controller selects the row-transforms as outlined in Fig. 2.

For fast inner-product implementation, we use a tree structure as shown in Fig. 3. Power savings are achieved by zeroing-out the input coefficients to the inner-product tree of Fig. 3. Here, we note that dynamic reconfiguration can be directly implemented using just 64 bits (one bit per DCT coefficient). However, to allow for variable number of bits for implementing the DCT coefficients, we need a DPR controller. For the purposes of this paper, we used 9 bits for $\pm a$ through $\pm g$ in (1), (2).

III. Results

The dynamically reconfigurable DCT architecture system was implemented on an XUP-ML505 reference board with a Virtex 5 FPGA device (xc5vlx110t). The 2D-DCT core depicted in Fig. 2 required 1710 slices at a maximum operating frequency of 182.4Mhz. A pipelined implementation of the 2D-DCT core required 16 cycles per 8×8 block.

The dynamically reconfigurable portion of the architecture is depicted in Fig. 4. The total bitstream size for this reconfiguration area is 29520 bytes, and it takes 5 frames.

In Fig. 5, for zig-zag encoding, we present estimated dynamic power consumption using Xilinx's Xpower Analyzer tool [5]. From the figure, it is clear that dynamic power increases linearly with the number of AC coefficients until we reach 30 coefficients. After 30 coefficients, dynamic power requirements flattens out. This is due to the fact that our zig-zag encoding requires the computation of DCT coefficients that are applied along each row and column (non-separable). We note that this limitation would not exist for rectangular encoding. As a function of the number of DCT coefficients, the dynamic power can be expressed using $\text{dpower} = 3.447 \times x + 110.9$ mW, for $x \in [1, 30]$, and $\text{dpower} = 215.7$ mW for $x \in [31, 64]$.

We tested the proposed system based on three sets of constraints given in Table I. Summary results for testing the entire LIVE image database are given in Fig. 6. From the results in Fig. 6, it is clear that the bitrate constraints are easilly met. Also, it is clear that there is limited variation in the dynamic power. It is clear from the graphs that image quality improves with bitrate and dynamic power.

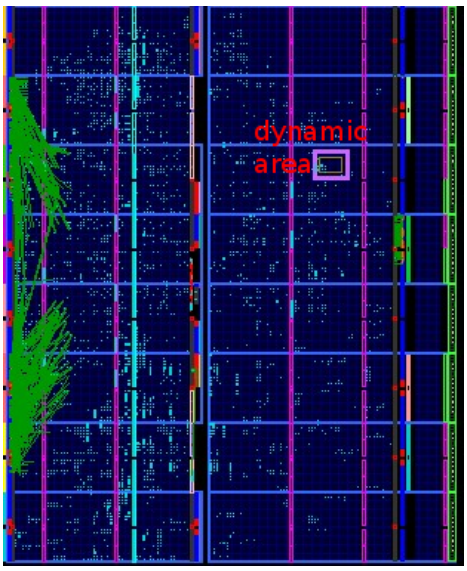

Figure 4. Hardware implementation of the entire system showing the dynamically reconfigurable (DR) region that includes the DCT zonal controller. Refer to Fig. 2 for the controller.

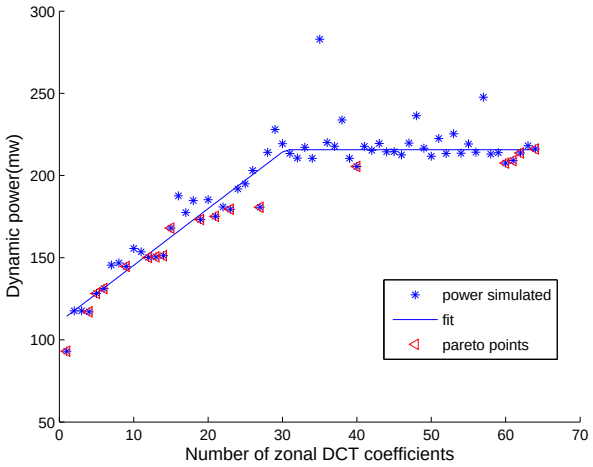

Figure 5. Dynamic power estimation based on the number of zig-zag encoded DCT coefficients. Note that dynamic power is not a function of the quality factor. The graph was generated for the Lena image.

Table I
DYNAMIC POWER AND BITRATE CONSTRAINTS USED IN TESTING (A, B, AND C). B_i REFERS TO BITRATE. DP_i REFERS TO DYNAMIC POWER.

	A: Low power & Low bitrate	**B**: Medium power & Medium bitrate	**C**: High power & High bitrate
B_i	0.5 bps	1.0 bps	1.5 bps
DP_i	160 mW	210 mW	230 mW

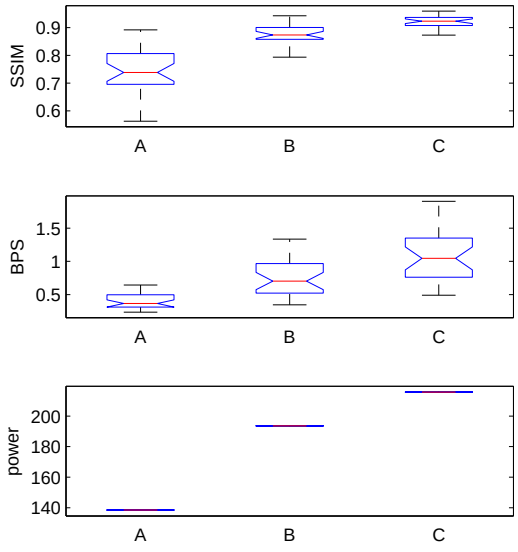

Figure 6. Performance testing boxplot results over the LIVE database (see Table I). Each LIVE image was used for testing while the rest was used for training. Constraint satisfaction results were 79.31% for A, 75.86% for B, and 86.21% for C.

We also present results for real-time reconfiguration based on the constraints $A \rightarrow B \rightarrow C$ of Table I. The performance of the optimal DCT architectures and corresponding quality factors are given in Fig. 7 and table II.

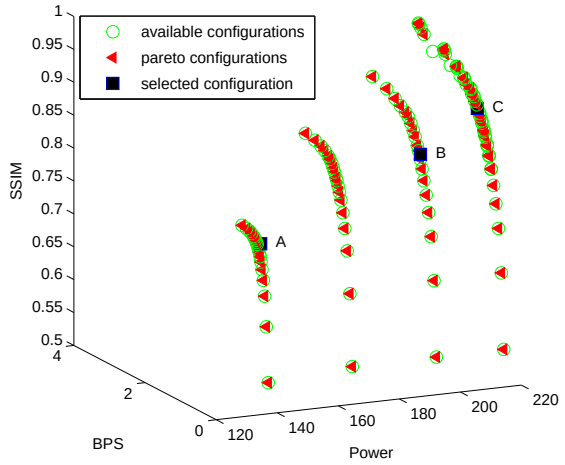

Figure 7. Multi-objective optimization results for A, B, C scenarios of Table I. Here, image quality is measured by SSIM, dynamic power is in milliWatts and bitrate is measured in bits per sample. The performance of the optimal DCT architectures and Quality factors are also plotted. The results are based on traning on the LIVE databases with the house image excluded (test image).

Table II
DYNAMIC RECONFIGURATION RESULTS FOR REAL-TIME CONSTRAINTS BASED ON A, B, C (SEE TABLE I) APPLIED TO THE HOUSE TEST IMAGE. THE REAL-TIME CONSTRAINTS ARE MET WITH SELECTED DCT ARCHITECTURES GIVEN BY $\mathcal{A}(N_i)$ WHERE N_i DENOTES THE NUMBER OF NON-ZERO DCT COEFFICIENTS.

	A ($i = 1$)	B ($i = 2$)	C ($i = 3$)
DCT Arch.	$\mathcal{A}(N_1 = 8)$	$\mathcal{A}(N_2 = 24)$	$\mathcal{A}(N_3 = 64)$
Quality Factor	$QF_1 = 40$	$QF_2 = 55$	$QF_3 = 75$
Bitrate	0.33 bps $\surd$	0.57 bps $\surd$	0.87 bps $\surd$
Dyn. Power	138.5 mW $\surd$	193.6 mW $\surd$	215.7 mW $\surd$
Im. Qual. (SSIM)	0.75	0.85	0.90

IV. CONCLUSIONS

In this paper, we present a dynamically reconfigurable DCT architecture system that can be used to maximize image quality while meeting real-time constraints on bitrate and dynamic power. Optimal DCT architectures are computed off-line and implemented in real-time using dynamic partial reconfiguration.

REFERENCES

[1] Wen-Hsiung Chen, C. Smith, and S. Fralick. A fast computational algorithm for the discrete cosine transform. *IEEE Transactions on Communications*, 25(9):1004 – 1009, sep 1977.

[2] L. Cormack H.R. Sheikh, Z.Wang and A.C. Bovik. Live image quality assessment database release 2.

[3] Cheng-Hsiung Hsieh. A zonal jpeg. In *International Conference on Information Technology, Coding and Computing, 2005. ITCC 2005.*, volume 2, pages 756 – 757 Vol. 2, april 2005.

[4] Jian Huang and Jooheung Lee. Reconfigurable architecture for zqdct using computational complexity prediction and bitstream relocation. *IEEE Embedded Systems Letters*, 3(1):1 –4, march 2011.

[5] Daniel Llamocca. *Dynamically Reconfigurable Management of Energy, Performance, and Accuracy Applied to Digital Signal, Image, and Video Processing Applications*. PhD thesis, University of New Mexico, 2012.

[6] John C. Hoffman Marios S. Pattichis. A high-speed dynamic partial reconfiguration controller using direct memory access through a multiport memory controller and overclocking with active feedback. *International Journal of Reconfigurable Computing*, 2011, 2011.

[7] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing,*, 13(4):600 –612, 2004.

[8] Xilinx Inc. *PlanAhead Software Tutorial Partial Reconfiguration of a Processor Peripheral*, september 2010.