Evaluating the Impact of Architectural Decisions on the Energy Efficiency of FDCT/IDCT Configurable IP Cores

Vinícius dos S. Livramento, Bruno G. Moraes, Brunno A. Machado, Eduardo Boabaid, and José Luís Güntzel

Department of Informatics and Statistics, Federal University of Santa Catarina, Florianópolis-SC, Brazil
e-mail: vini@inf.ufsc.br

ABSTRACT

The development of mobile multimedia devices follows the platform-based design methodology in which IP cores are the building blocks. In the context of mobile devices there is a concern of battery lifetime which leads to the need of energy-efficient IP cores. This paper presents four energy-efficient FDCT/IDCT configurable IP cores. These architectures are based on Massimino's algorithm, which was chosen due to its high accuracy and parallelism. The four architectures were built by combining fully-combinational or pipelined datapaths, using either a single or two 1-D DCT blocks with a transpose buffer that assures the optimal minimum latency of eight cycles. Synthesis results for 90nm showed that our most efficient architecture, which uses two pipelined 1-D blocks, achieved 250 MHz as maximum frequency at a total power of 14.03 mW. Such frequency was enough to process 16x 1080p@30fps videos in real time (nearly 2 GigaPixels/s). Comparisons with related work, in terms of energy efficiency (μJ/MPixels), revealed that our most energy-efficient architecture is at least 2 times as efficient as other DCT architectures. Moreover, the four designed architectures were also synthesized by using common low-power techniques. These results showed that pipelined versions at high throughput tend to take more benefit from using Low-Vdd and High-Vt combined than the combinational ones, thus becoming the most energy efficient.

Index Terms: Discrete Cosine Transform (DCT), VLSI architecture, Energy efficiency, low-power techniques.

1. INTRODUCTION

In recent years, battery-powered portable devices, such as digital cameras, tablets and smartphones, have driven the consumer electronics market. The design of such devices must take into account both performance and power consumption requirements. Particularly, power consumption must be kept as low as possible in order to prolong battery lifetime. Therefore, it is important to identify the main sources of energy consumption in battery-powered portable devices so as to concentrate the optimization effort in the most critical parts.

In current smartphone models baseband processing (e.g., 3G, Wi-Fi) is responsible for the largest amount of consumed energy [2]. In addition to baseband processing, internet browsing and media applications respond for another significant amount of energy consumption [3] [4], since they include coding/decoding pictures and videos, which are computationally intensive operations. Image and video coding and decoding follow multimedia standards, such as JPEG [5] and MPEG-1/2/4 [6]. Such standards make intensive use of several types of transforms being the Forward and Inverse Discrete Cosine Transform (FDCT/IDCT) [7] probably the most widely used ones.

The complexity of portable devices is increasing dramatically as more and more functionalities are incorporated each new generation. In order to cope with the ever increasing system complexity while satisfying the tight time-to-market, the development of integrated systems for mobile multimedia devices follows the platform-based design methodology [8], whose building blocks are Intellectual Property (IP) cores. Given a base platform, rapid prototyping is allowed by simply replacing and/or adding a couple of dedicated IP cores so as to customize the platform for the specific application requirements. Thanks to such powerful methodology the time-to-market requirements of complex systems-on-a-chip can be met.

Commercially available platforms, such as Texas Instrument's OMAP [9] and Qualcomm's Snapdragon [10] are mainly developed for application domains requiring high energy efficiency, such as mobile multimedia. Figure 1 shows the block diagram of OMAP DM5x platform, which is intended for digital cameras. The four main pro-

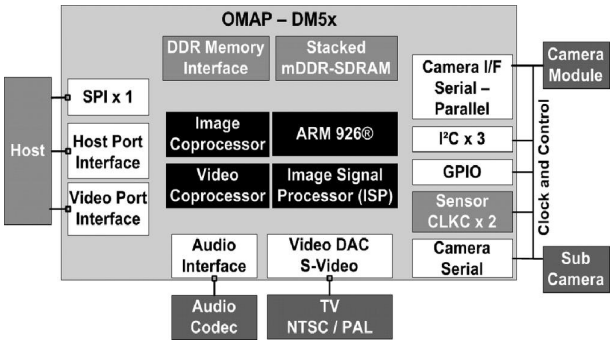

Figure 1. TI-OMAP DM5x platform (source: [1]).

cessing engines are highlighted as black boxes: an ARM926 general-purpose CPU, the Image Coprocessor (devoted to red eye reduction and backlight compensation processing), the Video Coprocessor (used for video stabilization and MPEG-4/H.264 encoding) and the Image Signal Processor (supporting JPEG compression at a maximum throughput of 90 MPixels/s - Mega pixels per second). This latter IP core is essential for keeping the energy consumption of the overall system within the required energy budget. In general, the use of dedicated IP cores tends to decrease the power consumption of the whole system because they offload the general purpose CPU by performing specific tasks more efficiently. However, the limited energy budgets make the power consumption issue still more stringent, in such a way that simply using IP cores may not be enough to meet low consumption requirements: the IP cores themselves must also consume the least amount of power. In addition, high energy efficiency may also be achieved by designing more specialized IP cores, able to compute often needed functions such as the Forward and Inverse Discrete Cosine Transform (FDCT/IDCT), as opposite to more general IP cores, as the Video Coprocessor in Figure 1.

In the design of energy-efficient IP cores the objective is to minimize the energy consumption while satisfying performance requirements. Therefore, the prime target is to optimize the "energy per operation", rather than optimizing the performance or the total energy consumption solely. The starting point in the design of an energy-efficient system is the estimation of the required performance. Considering the mobile devices that are developed using multimedia platforms (such as the one shown in Figure 1), smartphones and tablets correspond to a significant market share. Current generation of such devices code and decode color video up to a resolution of 1080p (1920 x 1080 pixels) at a frequency of 30fps (frames per second) in YCbCr [11] format with 4:2:2 subsampling. Under such format, a single frame in MPEG-2, for example, is represented by luma components (Y) and chroma components (Cb, Cr), each one organized as 8x8 pixel matrices as shown in Figure 2. To deal with such format in real time, a minimum throughput of 124.3 Mpixels/s is required. Obviously, such throughput allows for coding or decoding any other video

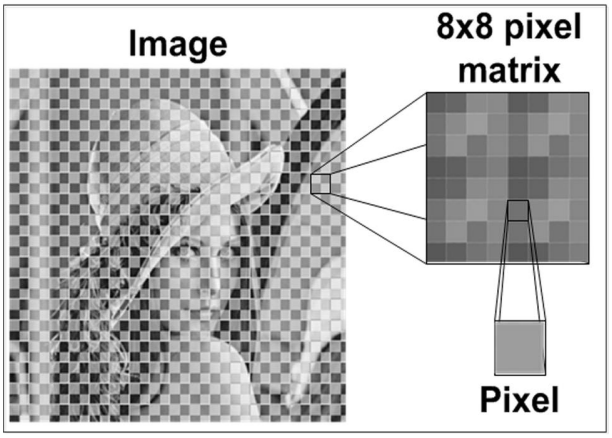

Figure 2. Image divided into 8x8 pixel matrices.

format having lower resolution or optionally, with higher subsampling. This includes, for instance, the VGA format (640x480) with 4:2:2 subsampling, which requires 18.4 Mpixels/s of throughput. These two examples of throughputs can serve as guidelines for designing and validating energy-efficient IP cores for multimedia applications.

This paper brings the following contributions: 1) We chose an accurate algorithm to perform the 8x8 2-D DCT to be used in JPEG, MPEG-1/2/4 and other compatible standards. We also modified the chosen algorithm to allow for resource reduction targeting performance and power optimization. 2) We designed and evaluated four different configurable architectures for the 8x8 2-D FDCT/IDCT based on the modified algorithm. The architectures use either a single or two 1-D block, which by they turn, can be fully-combinational or pipelined. Comparisons with a relevant set of related work revealed the high energy efficiency achieved by our four architectures. 3) Based on synthesis results, we provide indications on the most energy-efficient architecture for a given combination of throughput/technology node. 4) The four architectures were also synthesized by using low-power techniques. We show the impact of such techniques on the energy efficiency of each architecture along with its performance degradation.

This paper is organized as follows. The adopted 2-D DCT modified algorithm is presented in Section 2. Section 3 reviews the state-of-the-art, whereas Section 4 details the four designed architectures. Synthesis results and comparisons with related work are given in Section 5. Section 6 shows the synthesis results of the proposed architectures when low-power techniques are applied. Finally, Section 7 draws the paper conclusions.

2. THE ADOPTED 2-D DCT ALGORITHM

In order to facilitate the processing, image coding/decoding standards assume that each still image or frame (in the case of video) is divided into pixel matrices (sometimes referred to as macroblocks) with some allowed sizes. As an example, Figure 2 shows a still

image divided into 8x8 pixel matrices, which is adopted in JPEG standard. Thereby, the DCT computation is performed on each pixel matrix.

The 8x8 2-D DCT computation can be performed as a direct 2-D transform or as a sequence of three steps: 1-D DCT (performed on each 8-pixel row of the matrix) → transposition → 1-D DCT (performed on each 8-pixel column of the matrix). The latter approach explores the so-called separability property to reduce the 2-D DCT computational complexity, thus being widely employed in VLSI (Very-Large-Scale Integration) implementation.

For analysis purposes, row/column 1-D DCT dataflows are usually divided into two parts: one responsible for the even outputs (0, 4, 6, 2) and another one for the odd outputs (7, 5, 3, 1). The latter part is the most critical one concerning both accuracy and performance and therewith deserves special attention from designers. Some widespread solutions for the row/column decomposition are the Arai, Agui and Nakajima (AAN) [12] and the Loeffler, Ligtenberg and Moschytz (LLM) [13] algorithms. Figure 3 shows the AAN dataflow for an 8-pixel 1-D DCT, which is composed of 29 sums and 5 multiplications by constants. The LLM algorithm, on the other hand, requires 11 multiplications by constants and 29 sums to compute an 8-pixel 1-D DCT and presents high accuracy. Even though the AAN algorithm requires fewer multiplications, it has low accuracy in fixed-point arithmetic. Moreover, in the ANN datapath each odd output depends upon two consecutive multiplications (in Step 2 and Step 3, Figure 3), where the first one (in Step 2) is shared by the four even outputs. This characteristic makes parallelization more difficult and also decreases the algorithm accuracy due to numerical stability errors.

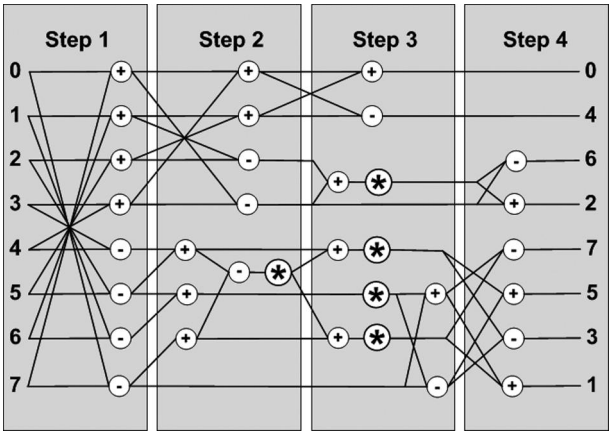

Figure 3. AAN FDCT dataflow composed of 29 sums and 5 multiplications.

Among the LLM-based approaches, Thomas Lane's (LibJPEG) [5] algorithm deserves to be highlighted since LibJPEG is a widely used JPEG implementation. It is based on an alternate and more balanced LLM dataflow that requires 12 multiplications by constants and 32 sums to compute an 8-pixel 1-D DCT. Figure 4 shows the LibJPEG dataflow which is composed of a single multiplication per odd path (i.e., no consecutive multiplications), hence being more accurate if compared to the AAN datapath, according to [14].

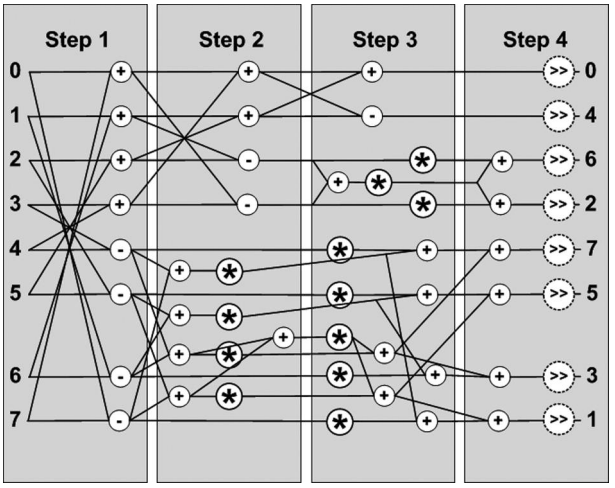

Figure 4. LibJPEG LLM-based FDCT dataflow composed of 32 sums and 12 multiplications.

The algorithm proposed by Pascal Massimino[1] [14], by its turn, is a fast and precise LibJPEG-based implementation characterized by its high accuracy in fixed-point arithmetic due to its cosine constants. Moreover, its dataflow was specially tailored to run in SIMD (Single Instruction, Multiple Data) architectures and therefore presents a well-balanced odd part. These features make this algorithm quite appropriate for VLSI implementations. Figure 5 shows the dataflow for an 8-pixel 1-D DCT, which is composed of 7 butterflies, 4 planar rotations (R6, R17, R13 and R37), 6 sums and 8 hardwired shifts to scale the results. As detailed by the set of equations 1, a planar rotation may be computed by 3 multiplications by cosine constants and 3 sums.

$$\begin{aligned} tmp &= (x + y).\cos(t) \\ x &= tmp + y.(\sin(t) - \cos(t)) \\ y &= tmp + x.(\sin(t) + \cos(t)) \end{aligned} \quad (1)$$

Due to the features mentioned in the previous paragraph, we have chosen Massiminos' algorithm to design four different architectures to compute the 8x8 2-D DCT.

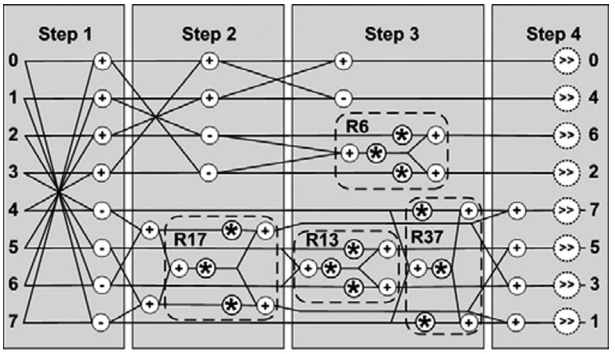

Figure 5. Massimino's original dataflow composed of 32 sums and 12 multiplications.

[1]Massimino's algorithm is licensed for academic purposes only.

To allow for fixed-point multiplications, Massimino employed a well-known technique to realize floating-point operations using only integer arithmetic, known as binary scaling. The binary scaling operation applied by Massimino resulted in 19-bit width integer cosine constants by using a binary point B16 (2^{16}). An example of binary scaling of a floating-point multiplication using a binary point B16 is presented through equations 2 to 6. The example shows a floating-point multiplication of 1.2 by 5.6 (equation 2). To obtain the fixed-point operands both numbers are multiplied by 2^{16} (in binary, accomplished by a simple shift left by 16), as shown in equations 3 and 4. As next steps, the fixed-point multiplication is performed (equation 5) and the result is divided by 2^{16} (in binary, accomplished by a simple shift right by 16). It is worth noting the high accuracy of such binary scaling multiplication, as shown in equation 6.

$$1.2 * 5.6 = 6.72 \quad (2)$$
$$1.2 * 2^{16} = 78643 \quad (3)$$
$$5.6 * 2^{16} = 367001 \quad (4)$$
$$78643 * 367001 = 28862059643 \quad (5)$$
$$28862059643/2^{16} = 6.7199 \quad (6)$$

A. Our Modifications on Massimino's Algorithm

As a first step to tailor Massimino's algorithm to design energy-efficient VLSI architectures, the original dataflow (Figure 5) was modified with the purpose of designing a configurable 1-D block, able to perform either the Forward (FDCT) or the Inverse DCT (IDCT). The Forward DCT of an 8-pixel row or column is accomplished by traversing Massimino's dataflow forwards (from left to right). Nevertheless, simply traversing such dataflow in backwards is not enough to compute the Inverse DCT. Therewith, the four steps of the original dataflow were rearranged into three steps (as shown in Figure 6) to enable the IDCT computation by traversing the dataflow in backwards (while the FDCT computation is still accomplished by a forward traversal). In summary, step 0 is composed of seven butterflies, step 1 is composed of four planar rotations and 6 sums, and step 2 is composed of some hardwired shifts.

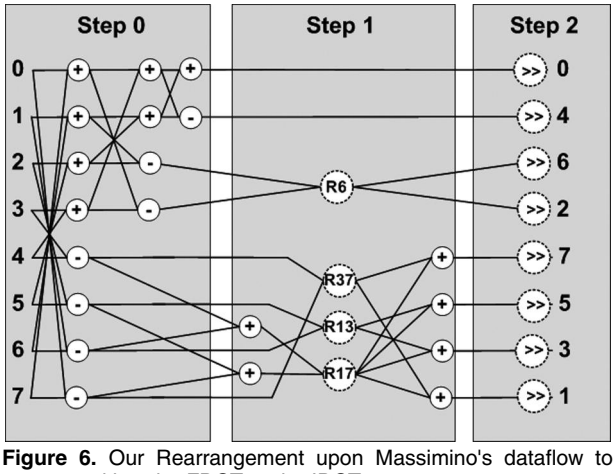

Figure 6. Our Rearrangement upon Massimino's dataflow to compute either the FDCT or the IDCT.

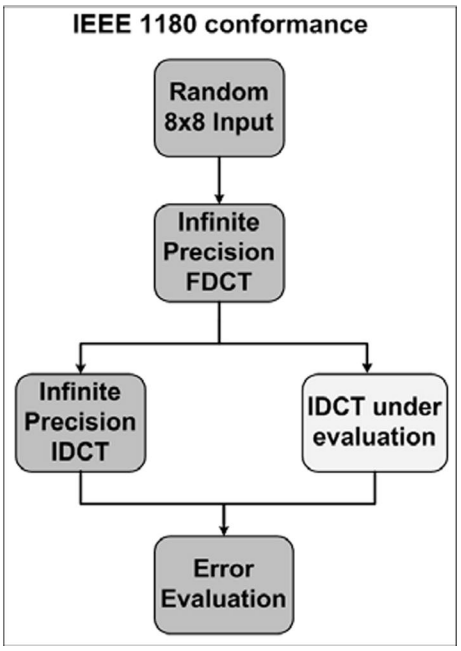

Figure 7. IEEE-1180 conformance test flowchart.

The IEEE-1180 conformance test [15] is a known metric to evaluate the IDCT precision. Figure 7 shows its test flowchart which consists basically of: a random 8x8 input matrix generator, an FDCT to generate the input coefficients, an infinite precision IDCT and an IDCT under evaluation. The error evaluation step verifies if the following error metrics are above the specified thresholds (defined within the conformance test): Peak Error (PE), Peak Mean Error (PME), Peak Mean Square Error (PMSE), Overall Mean Error (OME), and Overall Mean Square Error (OMSE). Table 1 shows these thresholds and also the results for Massimino's IDCT algorithm reported by the own author. The reported conformance values indicate the high precision achieved by Massimino's algorithm since it is below the specified thresholds for all error metrics.

Table 1. IEEE Conformance Results for Massimino's Algorithm.

Pixel Range	PE (<=1)	PME (<0.015)	PMSE (<0.06)	OME (<0.0015)	OMSE (<0.02)
(-256,255)	1.000	0.00191	0.0340	-0.00333	0.0200

Two additional important modifications were made in Massimino's algorithm looking to save resources for VLSI architectures. The first one was the change of binary point from B16 (2^{16}) to B11 (2^{11}) to reduce the integer cosine constants width and hence the multiplications width. The second one came from the clipping to 14-bit width (-8192 to 8191) of the row 1-D DCT outputs and consequently of the transpose buffer (responsible for transposing between row and columns). Both modifications were made by using the IEEE-1180 IDCT conformance test (Figure 7) as stop criterion. Figure 8 shows the IEEE-1180 errors for all tested binary-points and the IEEE-1180 thresholds. The Peak Error (PE) is 1 for all tested binary-points and hence it was omitted. The normalized dashed

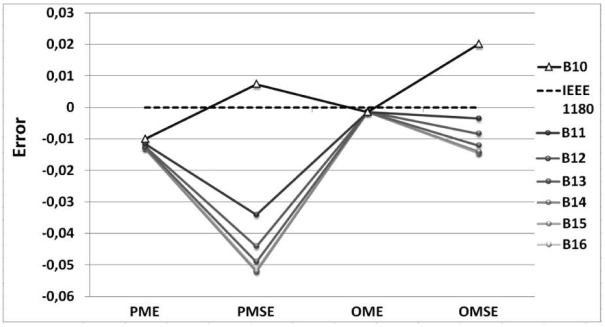

Figure 8. All tested binary-points.

line represents the threshold values for an IDCT to be compliant. It is important to mention that all X-axis dots are independent and the interpolation is just for a better understanding. As it can be observed in Figure 8, the binary-point B10 is not compliant since it is above the IEEE-1180 thresholds for PMSE and OMSE. Thus, our modified version of the algorithm uses B11 as binary-point to replace B16 in original Massimino's algorithm. The complete IEEE-1180 IDCT test of our modified version of the algorithm with the new binary-point (B11) and the clipping is shown in Table 2. It is worth noting that our modifications on Massimino's algorithm (change of binary point and clipping of the row 1-D DCT) reduced the precision of IEEE-1180 results when compared to the original algorithm (results shown in Table 1). Despite this precision reduction, our modified algorithm is still compliant with IEEE-1180 since it is below the specified thresholds.

Table 2. IEEE 1180 results for our modified version of Massimino's algorithm.

Pixel Range	Peak Error	PME (<0.015)	PMSE (< 0.06)	OME (<0.0015)	OMSE (<0.02)
(-300,300)	1	0.0035	0.0295	-0.0001547	0.0189641
(300,-300)	1	0.0032	0.0294	0.0003203	0.0189922
(255,-256)	1	0.0038	0.0327	0.0000797	0.0204578
(-256,255)	1	0.0041	0.0327	-0.0000156	0.0204500
(-5,5)	1	0.0025	0.0154	0.0002500	0.0129594
(5,-5)	1	0.0036	0.04	-0.0001859	0.0129859

Since the IEEE-1180 conformance test is mainly intended for the IDCT computation, we also used the popular objective quality metric called Peak Signal to Noise Ratio (PSNR) to test and compare the FDCT quality of three different algorithms: LibJPEG LLM-based implementation, original Massimino's algorithm, and our modified version of Massimino's algorithm. The PSNR flowchart is shown in Figure 9, and works as follows. In a first step, the three algorithms compute the FDCT for a PPM (Portable Pixel Map) input image. Then, the IJG (Independent JPEG Group) IDCT is used to compute the IDCT coefficients and thereafter generate the output PPM images for each algorithm. The three output images are compared to the original one. Table 3 shows the PSNR average results considering three well-known still pictures (*Lena*, *Peppers* and *Baboon*), each one in three different qualities. Such results, along with the IEEE-1180 test, indicate the high precision of Massimino's algorithm with our proposed modifications.

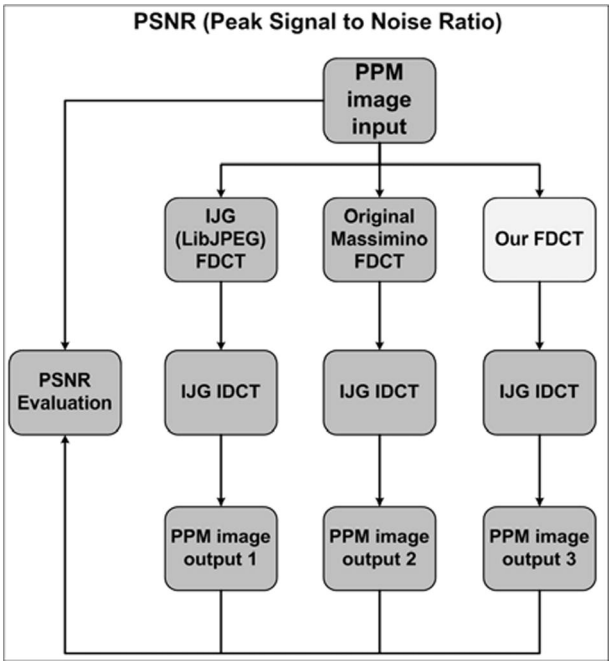

Figure 9. PSNR Test Flowchart.

Table 3. PSNR Results For Three Different Algorithms.

Quality [%]	IJG	Massimino's	Modified
50	32.950	32.943	32.943
75	35.153	35.140	35.143
100	54.033	54.150	54.143

3. REVIEW OF RELATED WORK

Energy-efficiency is a well-balanced compromise between power consumption and target performance, both coming from the application's requirements. Improving solely power or performance limits the achievable energy-efficiency. Most works improve performance by using pipelining along with the duplication of the 1-D DCT block. To cope with power issues, the main focuses of related work are the transpose buffer and multiplications within the DCT dataflow.

The transpose buffer is responsible for a large portion of the power consumption of a DCT computation, since it requires many memory elements. Sung [16] and Lee [17] used a direct 2-D transform which does not use a transposition structure but has a complex dataflow. Sung [16] uses subband decomposition of its dataflow along with factorized coefficient matrices in order to reduce computation complexity. The coefficients are computed, one by one, in a pipelined fashion. Lee [17], by its turn, uses a derived recursion equation to compute only non-zero elements. To explore the matrix sparseness, the following techniques are used: even/odd symmetry and zigzag scan order to also skip high probable zero coefficients. By doing so, the 8-point 2-D DCT is decomposed into two 4x4 matrices which are multiplied by 4x1 vectors in a pipelined manner. Hsia [18] eliminated the use of transpose buffer by adopting a particular

computing schedule based on a fast row/column 1-D DCT algorithm using two 1-D DCT blocks coupled with a temporal reorder buffer.

The specific implementation of the multipliers also impacts on power and performance. Taking advantage on constant coefficients used by the DCT, several techniques propose, as first optimization step, integer encodings that reduce multiplication costs. The Algebraic Integer Quantization (AIQ) technique is one of those. It defines roots of monic polynomials to represent irrational numbers in terms of integers. Such coding allows for virtually error-free arithmetic computations resulting in fast multiplierless parallel architectures. Wahid's [19] used two 1-D DCT blocks with a transpose buffer composed of two dual-port RAM memories. Its 1-D DCT block is organized as a linear multistage pipeline using 3-D algebraic integer encoding to map transcendental functions (e.g., cosine and sine). Fu's architecture [20] used two 1-D DCT blocks with 2-D AIQ and a dual port RAM as transpose buffer). The 3-D AIQ proposed by Wahid is sparser than Fu's 2-D AIQ and thus, requires fewer adders. Sung [21] uses two 1-D DCT blocks with an SRAM transpose buffer. Sung also reduces power consumption by implementing multipliers with Double-Rotation CORDIC algorithm in a circular coordinate system.

Another widely used technique to implement the multiplications is the distributed arithmetic approach. It speeds up the multiplication process by pre-computing all possible intermediate values and usually storing them in a ROM. Kim [22] and Chungan [23] reduced power by using distributed arithmetic with compressed adders in two 1-D DCT pipelined blocks and a single transpose buffer. Kim [22] arranges the minimal partial products of each DCT coefficient by using 2^k signed digit position to create custom compressed adders arranged in a Wallace tree. Chungan [23] saves power by using 15 adders to form a compressed adder tree which replaces the ROM to store coefficients.

A commonly used solution to implement multiplication by constant relies on decomposing it as shift-add operations. August [24] investigated the following low power techniques for the DCT: skipping 8x8 matrices when all elements are zero, skipping low energy 8x8 matrices, multiplications implemented as shift-add operations with low precision constants and clock-gated [25] registers that are update only when needed. Pai [26] exploited the LLM algorithm adding zero skip and truncation. Pai also compares different multiplier architectures and used the hardwired shift-add canonical sign-digit (CSD) Wallace-tree multiplier with segmented operands. Tsao [27], by its turn, reduced the number of shift-add CSD multipliers by using time rescaling.

Wahid presents two hybrid architectures targeting multiple standards. The first one [28] uses CSD while the second one [29] uses a multi-dimensional delta mapping.

Xanthopoulos [30] investigated the use of fine-grained clock-gating on Multiply-Accumulate units by exploring the occurrence of zero-valued coefficients. He also used pipelining to allow aggressive supply voltage scaling (Low-Vdd [25]) and high threshold voltage (High-Vt [25]) to reduce both dynamic and static power.

Unfortunately, the related works do not specify *a priori* the target application (and consequently do not make explicit the target throughput). Instead, they report the achieved throughput and power results only after synthesis. As a consequence, the architectures are not tailored for a specific application and therefore they may loose optimization opportunities. Besides the previously mentioned limitations, none of the related works investigated the impact of design decisions on the energy efficiency considering different application domains. This paper addresses this point by proposing and investigating the energy efficiency of four different architectures to compute the 8x8 FDCT/IDCT. In addition, it also points out the most appropriate application niche for each proposed architecture.

4. DESIGNED ARCHITECTURES

The intrinsic features of Massimino's algorithm, such as high degree of parallelism and integer cosine constants, are very appropriate for dedicated VLSI implementation. Thanks to the fixed-point cosine constants used by the algorithm, the multiplications are implemented as shift-add operations, one for each cosine constant. Such favorable features are still complemented by the cosine constants width reduction, as presented in Section 2, leading to a set of features that were explored to design fast and compact VLSI architectures.

With the objective of exploring different applications niche, we designed four dedicated 2-D FDCT/IDCT architectures based on our modified Massimino's algorithm, which processes 8 pixels in parallel. The architectures were derived by adopting either a single or two configurable 1-D DCT blocks. In a given implementation the 1-D DCT block was implemented either as a fully combinational or as a three-stage pipelined block, where the stages correspond to the dataflow steps shown in Figure 6. Besides the 1-D DCT block(s), each architecture has a transpose buffer (TBUFFER), composed of a register file organized as an 8x8 matrix of 14-bit width registers and a control unit. The latter is responsible for the input/output protocol (according to AMBA-AXI) and for the register file read/write operations.

The main features of the four architectures can be summarized as follows:

- **2xDCT Comb:** Uses two 1-D FDCT/IDCT fully-combinational blocks and a TBUFFER with a single write and a single read port[2].
- **2xDCT Pipe:** Uses two 1-D FDCT/IDCT three-stage pipelined blocks and a TBUFFER with a single write and a single read port.
- **1xDCT Comb:** Uses a single 1-D FDCT/IDCT

[2] In the TBUFFER, one port corresponds to eight 14-bit width data.

fully-combinational block and a TBUFFER with two write and two read ports.

- **1xDCT Pipe:** Uses a single 1-D FDCT/IDCT three-stage pipelined block and a TBUFFER with two write and two read ports.

A. 2xDCT_Combinational and 2xDCT_Pipeline

Figure 10 shows the 2xDCT Pipe block diagram. The Row 1-D FDCT/IDCT block is implemented according to the modified algorithm dataflow shown in Figure 6. The FDCT calculation is performed by the sequence STEP0, STEP1 and STEP2, whereas the IDCT is accomplished by the sequence STEP2, STEP1 and STEP0. STEP0 is composed of seven butterflies, STEP1 is built up from four rotates and some adders and STEP2 is composed of some hardwired shifters. Figure 11 shows the rotate block diagram, composed of three sums and three multiplications by constants. Each multiplication is implemented as shift-add operations, resulting in a total of twelve hardwired multipliers, one for each cosine constant, as commented in Section II. Each 8 x 14-bit width output data of row 1-D FDCT/IDCT is provided to the transpose buffer which, in turn, delivers it as inputs to the column 1-D FDCT/IDCT block. The column block slightly differs from the row block by some operations and constants that are required by the column computation. Figure 10 also shows the eight 12-bit input/output registers. The only difference between the fully-combinational and pipelined versions is the adoption (or not) of the pipeline registers inside the FDCT/IDCT blocks. It is worth mentioning that, although the use of pipeline results in an overhead (from registers and control), in our architectures such overhead is moderate since they use only three stages.

The transpose buffer featuring multiple read/write capabilities is detailed in Figure 12. It is responsible for transposing the received data from row to column order. The FSM describes the simultaneous read-write sequences, alternating by row/column order for each matrix. In State 0 it receives each one of the 8 x 14-bit width input data and writes them in rows A to G. State 1 writes the last input into row H and reads the column 0 bypassing the result from the last write. State 2 accomplishes the transposition by reading the columns 1 to 7 and delivering them as outputs. It also writes a new input matrix into columns 0 to 6. State 3 writes the last input into column 7 and reads the row A bypassing the previous write. State 4 is similar to State 2 except that it writes on columns and read from rows. The operation is repeated iterating through the States 0 to 4 until a last matrix is received (either in State 1 or 3). Thus, at maximum utilization this scheme allows the transpose buffer to achieve the lowest possible latency, which corresponds to eight cycles per matrix.

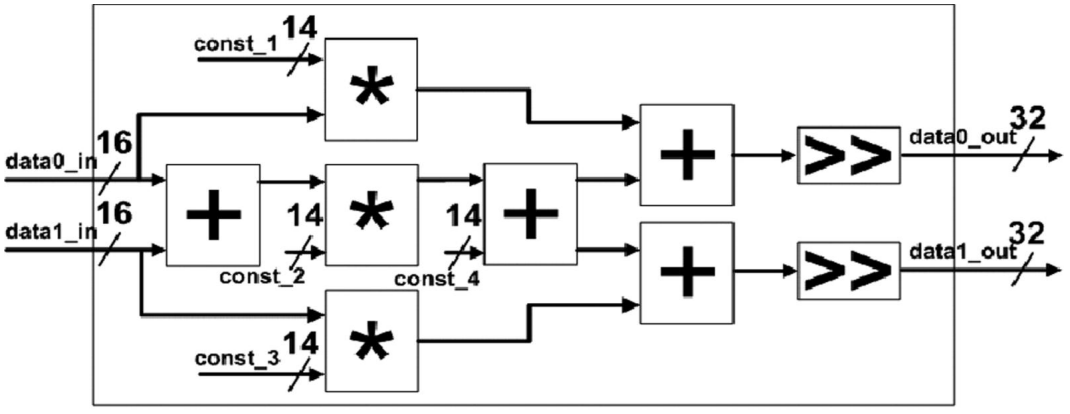

Figure 11. Rotate Block Diagram, used in STEP1.

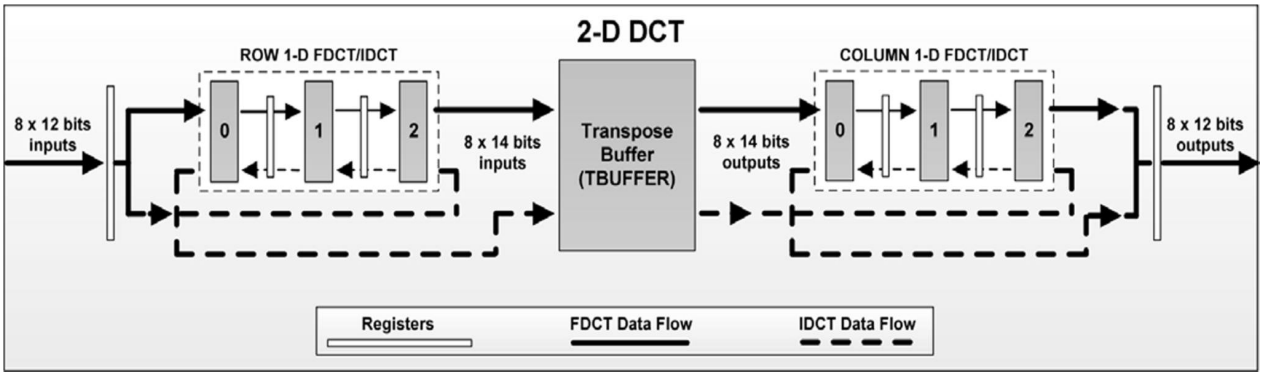

Figure 10. 2-D DCT with a two 1-D FDCT/IDCT Block

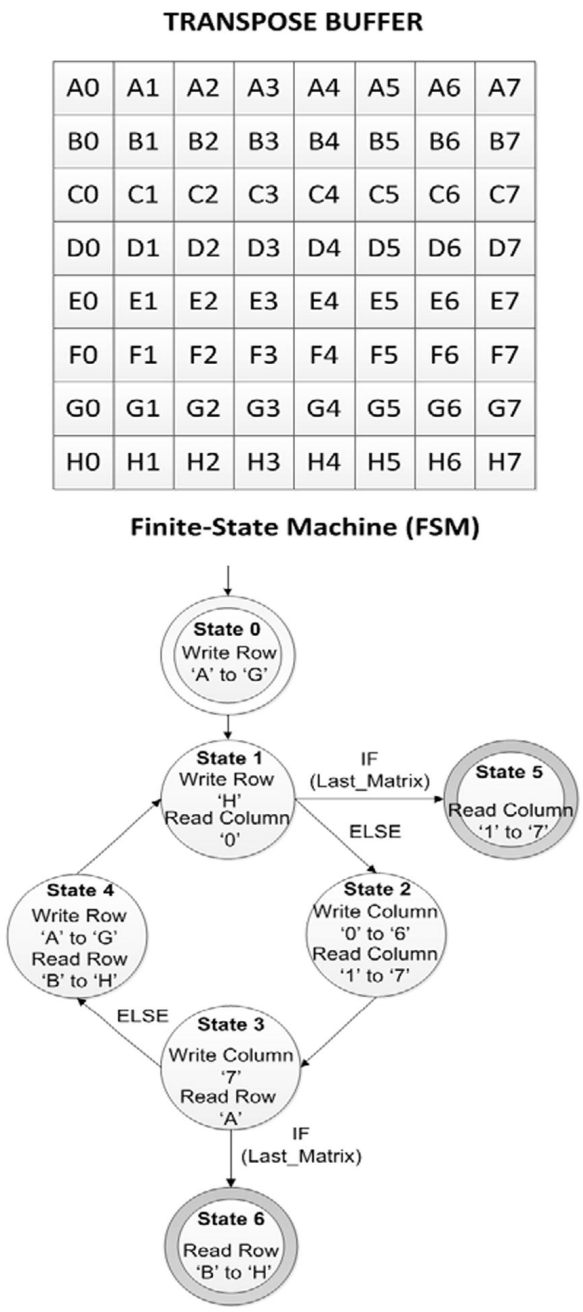

Figure 12. Transpose Buffer Operation for the 2xDCT Comb/Pipe.

B. 1xDCT_Combinational and 1xDCT_Pipeline

Figure 13 shows the 1xDCT Comb block diagram. In this case the transpose buffer is used as both input and output register to save resources. Hence its operation differs by first storing an input data in its register file, at the same time reading this data to the 1-D DCT block in row computation mode. The result is written back to the same position. This processing continues until the register file is filled. There is a one cycle delay to change the read/write transpose buffer sequence and to configure the 1-D block to column computation. After that, data is read as column, written back to the same position and dispatched as output. The adoption of a single 1-D DCT block requires seventeen cycles. The pipelined version needs four more cycles of latency to compute an entire matrix. For further details on the design of the configurable 1xD DCT Comb, refer to [31].

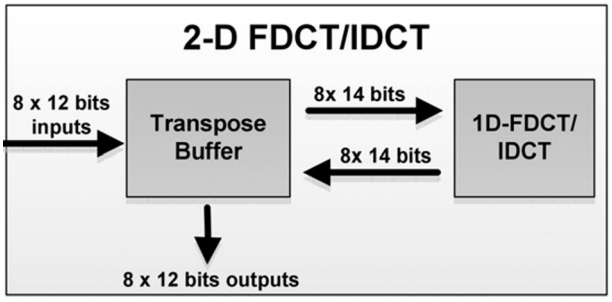

Figure 13. 2-D DCT with a single 1-D FDCT/IDCT Block.

C. Estimating the Required Clock Frequencies

Table 4 shows the resulting latencies for each of the four designed architectures. It also provides the required frequencies considering the two target throughputs estimated in Section 1: 124.3 MPixels/s for 1080p@30fps (4:2:2 subsampling) and 18.4 MPixels/s for VGA@30fps (4:2:2 subsampling). It is worth mentioning that in our previous work [31] the 1xDCT Comb architecture was synthesized and evaluated targeting a throughput of 93.3 MPixels/s for 1080p@30fps (4:2:0 subsampling).

To estimate the required frequency for each architecture to meet the throughputs, we took into account the latency of each architecture to compute an 8x8 matrix. The maximum frequency of each architecture, by its turn, is obtained only after synthesis. It is important to observe that the frequencies required for both 2xDCT architectures are less than half of those required by the 1xDCT architectures. This is because the 2xDCT architectures fully explored the multiple read and write capabilities provided by the transpose buffer by the addition of the second 1-D block.

Table 4. Latency and Target Frequencies for VGA and 1080p.

		Target Frequencies	
Architecture	**Latency**	**@18.4 MP/s (VGA)**	**@124.3 MP/s (1080p)**
1xDCT Comb	17 cycles	4.9 MHz	33.3 MHz
1xDCT Pipe	21 cycles	6.0 MHz	41.6 MHz
2xDCT Comb	8 cycles	2.3 MHz	15.6 MHz
2xDCT Pipe	8 cycles	2.3 MHz	15.6 MHz

5. SYNTHESIS RESULTS

The proposed 2-D DCT architectures were described in Verilog HDL and synthesized using Synopsys Design Compiler Topographical to obtain realistic post-layout timing, area and power estimates [32]. It is worth noting that neither dynamic power nor leakage power optimizations were enabled during the synthesis.

Synthesis results for the 90nm TSMC CMOS 1V standard cell library are summarized in Table 5. For

Table 5. Synthesis Results for the Four Proposed Architectures.

Architecture	Core Area [mm²]	Power@VGA (Dyn + Leak) [µW]	Power@1080p (Dyn + Leak) [µW]	Power@Max (Dyn + Leak) [µW]	Max. Freq. [MHz]	Throughput@Max [MPixels/s]
1xDCT Comb	0.107	148.8 + 193.8	1,003.8 + 192.9	1,534.0 + 201.5	50	188.3
1xDCT Pipe	0.111	180.3 + 190.4	1,240.4 + 190.2	6,042.5 + 205.6	200	609.6
2xDCT Comb	0.117	122.5 + 237.3	814.9 + 236.4	3,591.0 + 314.1	62.5	500
2xDCT Pipe	0.125	122.2 + 250.4	819.0 + 250.0	13,724.0 + 311.2	250	2000

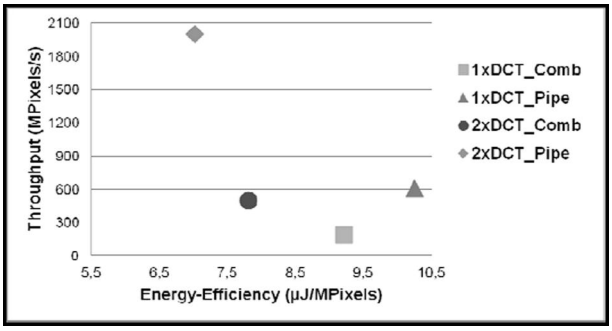

Figure 14. Throughput x Energy Efficiency at Maximum Frequencies.

Architecture it presents the core area, total power (Dynamic + Leakage) for the two target throughputs and their respective maximum achievable frequencies. The power values reported by Synopsys Design Compiler used random switching activity of 10% which is the default value used by the tool. Figure 14 presents the trade-off between the maximum achievable throughput and energy efficiency, defined as total power (µW) divided by the throughput (MPixels/s) and expressed in [µJ/MPixels], for the proposed architectures. It can be seen that the 2xDCT Pipe has the best trade-off and furthermore, its maximum throughput can achieve 2 GPixels/s (which is enough to process 16x1080p). The 2xDCT Comb presents 11.4% less efficiency and can achieve 500 MPixels/s (4x1080p). 1xDCT Comb and 1xDCT Pipe are 31.4% and 45.7% less energy efficient than 2xDCT Pipe, respectively. Table 5 shows, for each architecture, the following results reported by the Design Compiler: core area, total power at the required frequency for VGA and 1080p, total power and throughput at the maximum frequency. From Table 5, it follows that even though the 1xDCT Comb does not present the best energy efficiency at maximum throughput, it seems to be the best choice concerning an application with low throughput requirement (e.g., VGA). For an 1080p requirement, the 2xDCT Comb shows to be the best option concerning energy efficiency. Nevertheless, for 1080p the 1xDCT Pipe is a good choice case the target application needs low leakage requirements.

A. Comparison with related work

Table 6 shows the comparisons with relevant related work found in the literature. For each work, it shows the type of transform (FDCT, IDCT, HWT and DFT), technology node, reported throughput and total power.

Due to differences in architectures, such as degree of parallelism, pipeline depth, throughput and clock frequency, a direct comparison would not be fair. To allow a fair comparison, the energy efficiency metric along with normalization to 90nm node at 1.0V is used. For each related work, its reported total power is scaled from the original node/Vdd to 90nm/1.0V using formulas obtained from [24].

Different synthesis tools and methodologies are used by the related work to obtain their power (dynamic and leakage) results. Although different tools can result in different power estimations a direct comparison between different tools is commonly used in the literature, except on the cases when the technology nodes and Vdd are different. To address this last issue, we normalized the

Table 6. Energy Efficiency Comparisons.

Work	Type	Tech[nm]-(Vdd[V])	Throughput/ MPixels/s	Total Power [mW]	Scaled Energy Efficiency [µJ/mP]
Kim [22]	IDCT	600-3.3	400	900	417.3
Xanthopoulos [30]	IDCT	500-1.3	14	4.64	838.0
Sung2006 [21]	FDCT/IDCT	180-1.8	275.2	127.7	159.0
August [24]	IDCT	180-1.8	25.1	1.09	15.0
Sung2010 [16]	FDCT/IDCT	180-1.8	1000	102.2	35.0
Fu [20]	IDCT	180-1.8	75	7.5	34.2
Chungan [23]	FDCT	180-1.8	2000	121.26	20.7
Wahid2010 [28]	FDCT/DFT/HWT	180-1.8	100	15.38	52.7
Pai [26]	IDCT	180-1.6	320	9.56	14.5
Wahid2011 [29]	IDCT(JPEG,H.264,AVS,VC-1)	180-1.8	146	6.8	15.9
Wahid2007 [19]	FDCT	180-1.6	80	4.08	24.9
1xDCT Comb	FDCT/IDCT	90-1.0	188.3	1.73	9.2
1xDCT Pipe	FDCT/IDCT	90-1.0	609.6	6.24	10.2
2xDCT Comb	FDCT/IDCT	90-1.0	500	3.90	7.8
2xDCT Pipe	FDCT/IDCT	90-1.0	2000	14.03	7.0
Lee [17]	IDCT	90-1.2	14	0.556	23.0
Tsao [27]	IDCT	45-0.9	800	11.01	20.1

power values as mentioned in the previous paragraph. The following related work used Synopsys tools (Design Compiler or Prime Power) to obtain their power results: [21], [24], [16], [20], [23], [28], [26], [19], [27]. The power results reported by [30] are obtained by their own tool (similar to SPICE) using switching activities from 6 different MPEG videos. The tools used by [22], [29] and [17] to obtain the power values are not informed.

Among our designed architectures the most energy efficient one, considering maximum frequency, is the 2xDCT Pipe with 7.0 [µJ/MPixels]. In contrast, the 1xDCT Pipe is our least energy efficient architecture, being 45% worse than the 2xDCT Pipe. Therewith, the 2xDCT Pipe is used as reference to compare with related work.

Kim [22] proposed an IDCT architecture composed of two 1-D blocks and a transpose buffer. Its architecture uses a variable radix-2 technique to implement multiplications as shift-add. Our architecture is 59.6 times more energy efficient than [22].

Xanthopoulos [30] proposed an IDCT architecture that uses an adaptive bit-width technique and a row/column classification. It also uses two 1-D blocks and a transpose structure. Our architecture is 119 times more energy efficient than [30]. It is worth mentioning that 4.64 mW achieved by [30] is mainly due to the very low Vdd used (1.3V), which is below half of the nominal Vdd for a 500nm technology. This point is highlighted when we scale the power value to 90nm technology assuming nominal Vdd.

Sung [21] proposed a CORDIC-based FDCT/IDCT architecture, composed of two 1-D units, an SRAM for transposition and a coefficients ROM. Our architecture is 22.7 times more energy efficient than [21].

August [24] presented an IDCT architecture composed of one 1-D block and a transpose memory. It combines several low power techniques such as data skipping and registers gating. Our architecture is 2.1 times more energy efficient than [24].

Sung [16] proposed an FDCT/IDCT core based on sub-band decomposition algorithm. Its 1-D pipeline architecture uses a ROM to store the coefficients. Our architecture is 5 times more energy efficient than [16].

Fu [20] used an algebraic integer encoding to reduce power and its architecture is composed of two 1-D blocks and a transpose RAM memory. Our architecture is 4.8 times more energy efficient than [20].

Chungan [23] proposed a high throughput FDCT architecture based on distributed arithmetic scheme. It is composed of two 1-D blocks, designed as a five-stage pipeline and a transpose register array. Our architecture is 2.9 times more energy efficient than [23].

Wahid [28] proposed a hybrid architecture to compute FDCT, DFT (Discrete Fourier Transform) and HWT (Haar Wavelet Transform). Its 1-D processor is implemented as a configurable pipeline and its multiplications are designed as shift-adds. Our architecture is 7.5 times more energy efficient than [28]. Obviously it has a power overhead due to its configurable architecture. Even though, our improvement is quite significant, taking into account that our architecture performs FDCT or IDCT.

Pai [26] proposed an IDCT core based on data-dependent signal processing concept. It uses data bypassing, and hardwired multipliers to reduce power. Its architecture is composed of two 1-D blocks and a transpose matrix. Our architecture is 2 times more energy efficient than [26].

Wahid [29] proposed a resource sharing IDCT using multidimensional delta mapping to obtain the coefficients for JPEG, MPEG-2, H.264, VC-1 and AVS using only adders and shifters. Our architecture is 2.2 times more efficient than [29].

Wahid [19] used a 3-D algebraic integer encoding to minimize the quantization errors. Its FDCT architecture uses two 1-D pipeline blocks and a dual port transpose buffer. Our architecture is 3.5 times more energy efficient than [29].

Lee [17] proposed an IDCT architecture that explores the sparseness property of the coefficient matrix to reduce complexity and achieve a high throughput rate. It is implemented as direct 2-D transform and thus, no transposition is needed. Our architecture is 3.2 times more energy efficient than [17].

Tsao [27] investigates pipelined depths of two and six stages in a 1-D block. We used the reported dynamic power (leakage power is not reported by the author) for a 45nm node considering the 2-stage pipeline version. Our architecture is 2.8 times more energy efficient than [27].

The reported results put in evidence the importance of the algorithm choice as well as the appropriate architecture itself. Since the chosen algorithm has enough parallelism to provide high throughput, the four proposed architectures explored this feature combined with the transpose buffer instrinsic parallelism to minimize latency and reduce the power consumption. The synthesis results have also shown that the use of two 1-D DCT blocks allowed to better explore the transpose buffer latency and therefore achieved the better energy efficiency. Furthermore, the adoption of pipeline increased the throughput by four times and also increased the energy efficiency by 11% at maximum clock frequency. It is worth highlighting that even the 1xDCT Pipe is at least 1.4 times more energy efficient than the related work.

6. TOTAL POWER EVALUATION IN DEEP SUBMICRON NODES

This section aims to investigate the use of low-power techniques to further improve the energy efficiency of the proposed architectures. Table 7 shows the timing slacks from the synthesis for 90nm at a throughput of 124.3 Mpixels/s. In more recent technology nodes (e.g., 45nm), these slacks are even longer. Low power techniques such as low supply voltage (Low-Vdd) and High

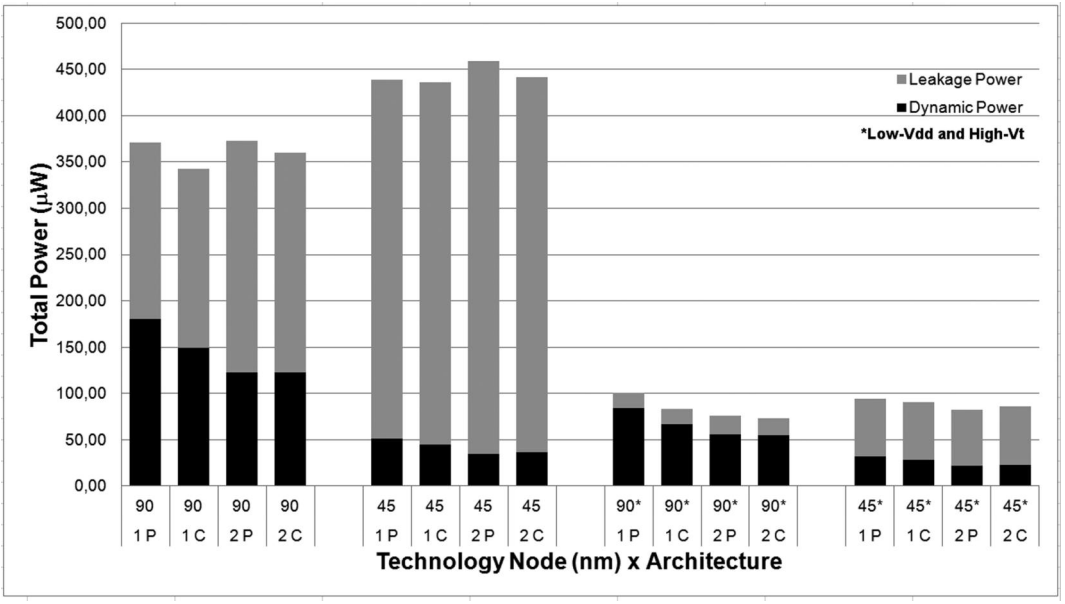

Figure 15. Evaluating the Use of Low-Vdd and High-Vt Using VGA (18.414 MPixels/s) as Target Throughput.

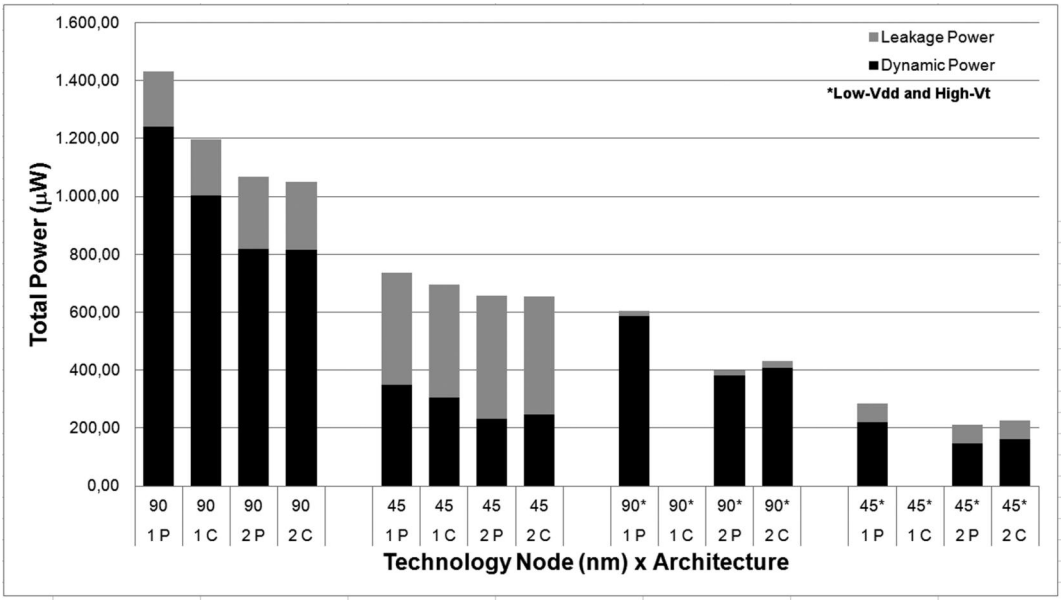

Figure 16. Evaluating the Use of Low-Vdd and High-Vt for Using 1080p (124.3 MPixels/s) as Target Throughput.

Table 7. Critical Path Slack For 90nm At 1080p.

Architecture	Slack (ns)
1xDCT Comb	14.10
1xDCT Pipe	26.61
2xDCT Comb	55.92
2xDCT Pipe	80.17

threshold voltage (High-Vt) [25] can be used to reduce the power consumption by exploring such timing slacks. In order to investigate the use of these techniques, the four proposed architectures were synthesized for 90nm (1V) and 45nm (0.9V) targeting VGA and 1080p for two scenarios:

1) Nominal Vdd/Vt;
2) Low-Vdd(0.7V)/High-Vt.

The graphics of Figures 15 and 16 show the total power consumption for both scenarios, at VGA and 1080p, respectively. In those graphics, the four architectures, 1xDCT Pipe, 1xDCT Comb, 2xDCT Pipe and 2xDCT Comb are referred to as, 1P, 1C, 2P and 2C, respectively. It is worth emphasizing that the same experimental setup used in Section 5 was used to obtain the power (dynamic and leakage) values. In Figure 15, one can note the leakage power dominance in Scenario 1, achieving 67% and 92% of total power for the 2P architecture in 90nm and 45nm, respectively. It is also important to notice that the total power consumption for 45nm nominal is, on average, 23% higher than the total power in 90nm nominal mainly due to leakage power dominance, consequence of the low clock frequency required by VGA.

$$degradation = 1 - \frac{maxFreq@LowVdd\_HighVt}{maxFreq@Nominal}$$

Figure 17. Maximum Frequency Degradation Formula.

Scenario 2 shows that the use of Low-Vdd/High-Vt can result in a total power reduction, on average, of 77% for 90nm and 80% for 45nm.

Concerning energy efficiency, the 1C architecture has shown to be the most energy-efficient option for 90nm and 45nm nominal (Scenario 1) targeting the VGA format. By using the low power techniques, the 2C architecture has shown to be the best option for 90nm, whereas the 2P architecture is the best option for 45nm.

By analyzing the graphic of Figure 16 in Scenario 1, one can observe the dynamic power dominance in 90nm, achieving 86.7% of the total power for the 1P architecture. On the other hand, in 45nm the leakage accounts for the most significant part, reaching 64.6% of the total power for the 2P architecture. In the case of Scenario 2, one can observe that the 1C architecture does not achieve the target throughput. The use of Low-Vdd/High-Vt leads to a total power reduction, on average, of 40.2% for 90nm and 35% for 45nm.

Concerning energy efficiency, the 2C architecture has shown to be the most energy-efficient option for 90nm and 45nm nominal (Scenario 1) targeting the 1080p format. By using the low power techniques, the 2P architecture has shown to be the best option for both 90nm and 45nm.

The use of low power techniques, such as Low-Vdd and High-Vt on all circuit gates obviously degrades the performance. Figure 18 shows such degradation for the four proposed architectures synthesized in 90nm and 45nm for the target throughput of 124.3 MPixels/s (1080p). The performance degradation with respect to the maximum frequency is calculated as shown in Figure 17. Figure 18 presents the energy efficiency improvement of the four architectures with respect to the nominal cases. Even though the performance degradation can achieve up to 82% (2P@45nm), most of the architectures meet the required throughput, except for 1C in 90nm and 45nm in which the use of low power techniques jeopardized the required clock frequency of 33.3MHz (shown in Table 4). For all other architectures (1P, 2P and 2C) in both 90nm and 45nm the required throughput was achieved with up to 4.05 times energy efficiency improvements. Clearly specifying the target throughput is essential to allow for a broader exploration of the design space to find the best combination between architecture and low-power techniques.

7. CONCLUSIONS

This paper presented four 2-D FDCT/IDCT configurable architectures based on Massimino's algorithm that features high degree of parallelism and high accuracy. Further modifications made in the algorithm, such as the adoption of binary-point B11 and row data 14-bit clipping, allowed for resource savings without compromising its accuracy, as verified by the IEEE 1180 IDCT conformance tests and PSNR tests.

The four architectures explored the separability property to process the 2-D DCT into two 1-D passes. Particularly, two architectures use two 1-D blocks whereas the others reuse a single 1-D block. In addition, the 1-D blocks are either fully combinational or pipelined. The designed transpose buffer with simultaneous/multiple read and write capabilities is a key component in the four architectures, since it assures the optimal minimum latency of eight cycles, thus greatly contributing to achieve high efficiency.

Synthesis results reported by Synopsys Design Compiler Topographical for 90nm 1V nominal TSMC standard-cell library, for maximum frequency, showed that our four architectures are between 2 and 147 times as energy efficient as relevant related work.

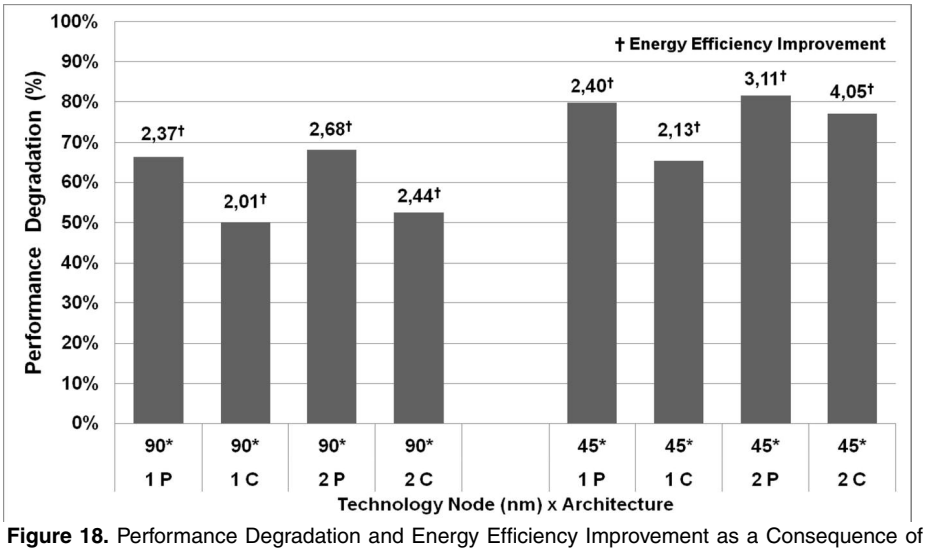

Figure 18. Performance Degradation and Energy Efficiency Improvement as a Consequence of Using Low-Vdd and High-Vt.

The four designed architectures were also synthesized for 45nm nominal, as well as for 90nm and 45nm by using low-Vdd/high-Vt by assuming VGA and 1080p as target throughputs. For VGA throughput the 1xDCT Comb is the most efficient architecture for both nominal nodes. However, when using low-Vdd/high-Vt the most energy efficient architectures are the 2xDCT Comb for 90nm and the 2xDCT Pipe for 45nm. Concerning the 1080p throughput the 2xDCT Comb is the most energy efficient architecture for both nominal nodes. With low power techniques the most energy efficient architecture is the 2xDCT Pipe, which is capable to process 2 GPixels/s. It is worth noting that the 1xDCT Comb was not able to fulfill the target throughput. Clearly, the combinational architectures have shown to be the most energy efficient architectures for all nominal cases. Finally, by applying Low-Vdd/High-Vt techniques the 2xDCT Pipe is the most energy-efficient at 45nm, which is not true for 90nm at the VGA throughput, when the 2xDCT Comb becomes the most efficient.

ACKNOWLEDGEMENTS

This work was partially supported by the Brazilian Council for Scientific and Technological Development (CNPq), through the "Brazil-IP Project". The authors also acknowledge the anonymous reviewers for their constructive comments.

REFERENCES

[1] T. Instruments. TI OMAP-DM5x. [Online]. Available: http://goo.gl/3rgIK

[2] A. Pathak, Y. C. Hu, and M. Zhang, "Where is the energy spent inside my app? fine grained energy accounting on smartphones with Eprof," in *Proceedings of the 7th ACM European Conference on Computer Systems*, *(EuroSys),* ACM, 2012.

[3] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan,and D. Estrin, "Diversity in smartphone usage," *in Proceedings of the ACM/USENIX International Conference on Mobile Systems, Applications, and Services (MobiSys),* ACM, June 2010.

[4] C. Shepard, A. Rahmati, C. Tossel, L. Zhong, and P. Kortum, "Livelab: measuring wireless networks and smartphone users in the field," in *Proceedings of the ACM Workshop on Hot Topics in Measurement & Modeling of Computer Systems (HotMetrics)*, ACM, June 2010.

[5] I. J. Group. Home site. [Online]. Available: http://www.ijg.org

[6] M. P. E. G. (MPEG). Home site. [Online]. Available: http://www.mpeg.org/

[7] N. Ahmed, T. Natarajan, and K. Rao, "Discrete cosine transform," *IEEE Transactions on Computers,* vol. 100, no. 1, 1974, pp. 90–93.

[8] K. Keutzer, A. Newton, J. Rabaey, and A. Sangiovanni-Vincentelli, "System-level design: Orthogonalization of concerns and platform-based design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.* vol. 19, no. 12, 2000, pp. 1523–1543.

[9] Texas instruments. OMAP. [Online]. Available: http://www.ti.com/omap/

[10] Qualcomm. Snapdragon. [Online]. Available: http://www.qualcomm.com/snapdragon/

[11] J. Murray and W. VanRyper, "*Encyclopedia of graphics file formats,*" Sebastopol: O'Reilly,— c1996, 2nd ed., vol. 1, p. 1116, 1996.

[12] Y. Arai, T. Agui, and M. Nakajima, "A fast DCT-SQ scheme for images," *IEICE Transactions*, vol. 1976, 1988, pp. 1095–1097.

[13] C. Loeffler, A. Ligtenberg, and G. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in *Proceedings of* the IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP), 1989, pp. 988–991.

[14] P. Massimino. DCT implementation site. [Online]. Available: http://web.archive.org/web/20060707003114/skal.planet-d.net/coding/dct.html

[15] I. C. S. Committee et al., "*IEEE standard specifications for the imple-mentations of 8x8 inverse discrete cosine transform*," IEEE standard, pp. 1180–1190, 1990.

[16] T. Sung, Y. Shieh, and H. Hsin, "An efficient VLSI linear array for DCT/IDCT using subband decomposition algorithm," *Mathematical Problems in Engineering*, vol. 2010, pp. 1–21, 2010.

[17] J. Lee, N. Vijaykrishnan, and M. Irwin, "Inverse discrete cosine transform architecture exploiting sparseness and symmetry properties," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 5, 2006, pp. 655–662.

[18] S. Hsia, C. Tsai, S. Wang, and K. Hung, "Transposed-memory free implementation for cost-effective 2D-DCT processor," *Journal of Signal Processing Systems*, vol. 58, no. 2, 2010, pp. 161–172.

[19] K. Wahid, V. Dimitrov, and G. Jullien, "On the error-free realization of a scaled DCT algorithm and its VLSI implementation," *IEEE Transactions on Circuits and Systems II*, vol. 54, no. 8, 2007, pp. 700–704.

[20] M. Fu, G. Jullien, V. Dimitrov, and M. Ahmadi, "A low-power DCT IP core based on 2D algebraic integer encoding," in *Proceedings of the International Symposium on Circuits and Systems* (ISCAS), 2004, vol. 2., pp. II–765.

[21] T. Sung, M. Sun, Y. Shieh, and H. Hsin, "Memory-efficiency and high-speed architectures for forward and inverse DCT with multiplierless operation," *Advances in Image and Video Technology*, pp. 802–811, 2006.

[22] D. Kim and J. Choi, "Variable radix-2 multibit coding for 400 mpixel/s DCT/IDCT of HDTV video decoder," *Integration, the VLSI Journal*, vol. 35, no. 2, 2003, pp. 47–67.

[23] P. Chungan, C. Xixin, Y. Dunshan, and Z. Xing, "A 250MHz optimized distributed architecture of 2D 8x8 DCT," in *Proceedings of the IEEE International Conference on ASIC, (ASICON), 2007,* pp. 189–192.

[24] N. August and D. Ha, "Low power design of DCT and IDCT for low bit rate video codecs," *IEEE Transactions on Multimedia*, vol. 6, no. 3, pp. 414–422, 2004.

[25] M. Keating, D. Flynn, R. Aitken, and K. Shi, *Low power methodology manual: for system-on-chip design*. Springer Verlag, 2007.

[26] C. Pai, W. Lynch, and A. Al-Khalili, "Low-power data-dependent 8×8 DCT/IDCT for video compression," *IEE Proceedings - Vision, Image and Signal Processing*, vol. 150, no. 4., IET, 2003, pp. 245–255.

[27] Y. Tsao and K. Choi, "Cost reduction on high-speed 1D IDCT architecture based on time rescaling," in *Proceedings of the IEEE International Conference on in Electro/Information Technology (EIT)*, 2011, pp. 1–6.

[28] K. Wahid, M. Islam, S. Shimu, M. Lee, and S. Ko, "Hybrid architectureand VLSI implementation of the cosine–fourier–haar transforms," *Circuits, Systems, and Signal Processing*, vol. 29, no. 6, 2010, pp. 1193–1205.

[29] K. Wahid, M. Martuza, M. Das, and C. McCrosky, "Resource shared architecture of multiple transforms for multiple video codecs," in *Proceedings of the 24th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2011, pp. 947-950.

[30] T. Xanthopoulos and A. Chandrakasan, "A low-power IDCT macrocell for MPEG-2 mp@ml exploiting data distribution properties for minimal activity,",*IEEE Journal of Solid-State Circuits*, vol. 34, no. 5, 1999, pp. 693– 703.

[31] V. Livramento, B. Moraes, B. Machado, and J. Güntzel, "An energy- efficient FDCT/IDCT configurable IP core for mobile multimedia platforms," in *Proceedings of the 24th Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2011, ACM, pp. 149–154.

[32] Synopsys, "Design compiler user guide," 2009.