A HIGH THROUGHPUT MULTIPLIERLESS LOW POWER 8X8 2-D DCT IP FOR PORTABLE MULTIMEDIA APPLICATIONS

Vinícius dos S. Livramento, Bruno G. Moraes, Brunno A. Machado, José Luís Güntzel
{vini, brunogm, guntzel}@inf.ufsc.br
brunno@das.ufsc.br
Federal University of Santa Catarina
Department of Informatics and Statistics

ABSTRACT

This paper presents a high throughput multiplierless low power IP core for the 8x8 2-D DCT. It relies on a fast and precise implementation of the LLM algorithm. The IP architecture explores the separability property, using a single 1-D DCT, implemented as a combinational block, an 8x8 register file and a control unit. The 1-D DCT block can operate at 5 MHz thus achieving a throughput of 19 Mpixels/s (VGA@30fps), as required by a typical contemporary portable multimedia application. Synthesis results for the XFab 350nm CMOS technology estimate total power as 7.65mW and core chip area as 3.20mm^2. In order to allow the comparison with related works, we used the mJ/Mpixels metric. Under such metric, the designed IP is at least twice as efficient as other 2-D DCT architectures presented in related works.

1. INTRODUCTION

Multimedia applications are present in most of contemporary portable electronic devices, such as smartphones and digital cameras. Such kind of applications demands high computational performance that, by its turn, results in significant power consumption, shortening battery lifetime. Image and video applications are heavily based on the Discrete Cosine Transform (DCT) [1], which is widely used in JPEG [2] and MPEG2 [3] multimedia coding systems. The high computational cost of the DCT and the demanded energy efficiency for battery-operated portable devices motivate the investigation of low-power DCT hardware architectures.

A contemporary smartphone has a built-in camera with up to 8 Mpixels of resolution, being capable to process a color still picture in YCbCr [4] with 4:2:2 subsampling format. Under such format, a single picture is represented by luma components (Y) and chroma components (Cb, Cr), each one organized as 8x8 8-bit pixel matrices. In the present paper the throughput is defined in Mega pixels per second and includes color information from the image. Therefore, for the smartphone mentioned above, which may be considered as a potential target application, the required throughput for a still picture is 16 Mpixels/s.

This work is partially supported by CNPq Brazilian Agency, through the Brazil IP project.

For the same target application, the throughput for the VGA@30fps video format is 18.432 Mpixels/s. Assuming this throughput, Table 1 shows other achievable video formats.

TABLE I. ACHIEVABLE VIDEO FORMATS WITH 18.432 MPIXELS/S

	VGA (640x480)	**30 fps**
Frame Rate 4:2:2	480x360	53 fps
	CIF (352x288)	90 fps
	QVGA (320x240)	120 fps

Many algorithmic approaches for the 8x8 2-D DCT can be found in the literature. The two most used solutions are the direct 2-D transform and the row/column decomposition, which computes the 2-D DCT as a sequence of two 1-D DCT computations and a transposition. Such an approximation is possible thanks to the so-called separability property of the 2-D DCT. This work adopts the latter solution since it leads to lower computational complexity, which may be explored to minimize the cost of the hardware. In addition, the lower computational complexity leaves room for exploring the tradeoff between performance and power consumption, by low power architectural exploration, possibly applying low power techniques such as Multi-Vdd and Multi-Vt [14].

A few fast 1-D DCT algorithms, such as the LLM [5] and the AAN [6], use butterflies processing and even/odd decomposition to reduce redundant computation.

Among the architectural solutions found in the literature, the 2-D DCT architectures proposed by Hsia [7], by Kinane [8] and by Agostini [9] are the ones that are nearest to the features presented by the architecture of this work. Therefore, they were selected to establish the comparisons.

The 2-D DCT proposed by Hsia [7] uses a low cost transposeless 1-D scheme with a particular scheduling order whose objective is to eliminate the need for a transpose operation between the two 1-D blocks. This is accomplished by using an off-chip memory. Kinane's solution [8] uses the 1-D scheme with a transpose memory using even/odd decomposition and a distributed arithmetic adder summation tree. This architecture also presents a shape adaptive structure with the DCT. The architecture presented by Agostini [9] is composed of two 1-D DCT blocks and a transpose buffer in-between. The multiplications are decomposed into shift-add operations.

Our approach is based on Massimino's[1] fast and precise LLM algorithm [10]. It performs two 1-D DCT computations that operate over rows and columns, respectively. In our implementation, a few particular architectural decisions were taken so as to minimize power consumption.

This paper is organized as follows. The 2-D DCT algorithm is presented in Section 2. The implemented architecture is detailed in Section 3. Synthesis results are shown in Section 4. Section 5 draws some conclusions.

2. THE CHOSEN 2-D DCT ALGORITHM

Table 2 shows the IEEE 1180 conformance [13] results for Massimino's 2-D DCT algorithm, reported by the own author. In this table, PE, PME, PMSE, OME and OMSE stand for peak error, peak mean error, peak mean square error, overall mean error, and overall mean square error, respectively. The reported conformance values indicate the high precision, which is one of the main reasons for choosing this algorithm for a hardware realization.

Considering 8x8 matrices, as used in the architecture presented in section 3, the complexity of Massimino's algorithm is **O**(N*N.lnN), where N is the DCT size.

Massimino's algorithm computes the 2-D DCT by using the row/column decomposition, that is, by two subsequent 1-D DCT computations and a transposition. Each 1-D computation is divided in two main parts: the first one realizes the initial butterflies, whereas the second one corresponds to the even/odd decomposition with four rotations, totalizing twelve multiplications. The rotations are decomposed using the following equations:

$$tmp = (x+y).\cos t \quad (1)$$
$$x' = tmp + y.(\sin t - \cos t) \quad (2)$$
$$y' = tmp - x.(\sin t + \cos t) \quad (3)$$

Also, the rotations were further reduced by common sub-terms factorization using the following trigonometric identity for the special case b=4.π/16:

$$\cos(a)*\cos(b) - \sin(a)*\sin(b) = \cos(a+b) \quad (4)$$

Each rotation has three multiplications, totalizing twelve multiplications for each 1-D computation. The cosine constants used in the multiplications are integers. That is another reason for choosing Massimino's algorithm. After all simplifications are done, the algorithm ends up with four rotations, some additions and a row/column decomposition. Those operations are quite simple and thus appropriate for direct hardware implementations.

3. 2-D DCT PROPOSED ARCHITECTURE

Originally, the algorithm performs the operations using nineteen-bit width constants. However, in order to reduce the amount of resources needed to implement it directly on hardware we recalculated the values of the constants in such a way that each constant is represented with fourteen bits. The new version of the algorithm incorporating such modifications was also submitted to the IEEE 1180 conformance tests. The results showed an error smaller than 0.01% for the OME and an error of 20% for OMSE with respect to the original version, whereas PME and PMSE did not change. Such results indicate a negligible loss of precision, despite the significant reduction in the resources achieved.

TABLE II. IEEE CONFORMANCE RESULTS FOR MASSIMINO'S 2-D DCT ALGORITHM

Input pixel range	PE (<=1)	PME (<0.015)	PMSE (<0.06)	OME (<0.0015)	OMSE (<0.02)
-256 ~ 255	1.0000	0.0191	0.0340	-0.0033	0.0200

Due to the high cost of hardware multipliers and considering that only multiplications by constants are needed, the proposed architecture adopts only shift-add operations (as does Agostini's [9]). This hardware optimization does not imply in any precision loss, since the constants used in Massimino's algorithm are integers.

The design of the 2-D DCT architecture aimed at a throughput of 19 Mpixels/s, considering that a contemporary portable multimedia device needs 18,432 Mpixels/s to process video in VGA@30fps with 4:2:2 format and 16 Mpixels/s to process an 8 Mpixel still image with 4:2:2 format.

The design also takes advantage of the separability property by using a single 1-D DCT block. A transpose buffer, built up from a register file and a control unit (FSM), supplies the 1-D DCT block with the 14-bit data in the appropriate order, which is first in row order and second in column order. The resulting matrix is given in row order, unlike related works [7][8][9], thereby improving the IP's reusability, since it is not necessary to modify the quantization stage to correct the output coefficients. Another advantage is that the designed architecture delivers the output coefficients as non-scaled output values, and therefore no further scaling is needed.

Figure 1 shows the block diagram of the designed 2-D DCT IP core. Its main blocks are the 1-D DCT and the transpose buffer (TBUFFER). The former is a combinational block. The latter is composed of a control and a sixty-four 14-bit register file, organized as an 8x8 matrix. The IP core interface consists of eight 8-bit data inputs, eight 12-bit data outputs, and control signals to perform the handshake according to the AMB-AXI protocol [13].

As a first step, the TBUFFER receives data from the data inputs, storing them in the registers in a row order. Then it feeds the 1-D DCT block with data stored in the registers following the row order and commands the storage of the results back to the registers, also in row order. In a third step, the TBUFFER feeds the 1-D DCT block with data stored in the registers following the

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

column order and commands the storage of the results back to the registers, also in column order. Finally, the resulting matrix is released by a row order read of the registers.

The parallelism provided by the architecture allows an 8x8 matrix to be read in 8 clock cycles. At the same time the data is read in it is processed in the 1-D DCT block. At the end of each cycle the results from the 1-D DCT are written back to the register file. Therefore, the row order 1-D DCT computation is accomplished in 8 clock cycles. More eight cycles are needed to perform the column order 1-D DCT computation. Finally, the whole resulting matrix is delivered in row order in another eight cycles. A straightforward optimization relies on not waiting until the whole resulting matrix to be delivered in order to begin the processing of a new matrix. At the same time the resulting matrix is being released, a new matrix begin to be processed, improving the throughput.

Figure 2 shows the IP timing diagram. Note that the latency is twenty-four cycles whereas the time to deliver a new matrix is seventeen cycles, considering steady operation. Such processing time leads to an estimated throughput of 19 Mpixels/s when operating at 5MHz.

The 1-D DCT block was designed as a fully combinational block. Such design decision was motivated by the following reasons:

- The level of parallelism used in the architecture makes possible to compute the 1-D DCT of a whole row or column within a single clock cycle, thus avoiding the cost and power consumption of a pipeline processing.
- The control unit is simple.
- Lower clock frequency as compared to a pipeline version leading to lower switching activity.
- The combinational 1-D DCT leads to a clock slack that is longer than that of a pipelined structure, opening the possibility to apply advanced low power techniques such Multi-Vdd and Multi-Vt [14].

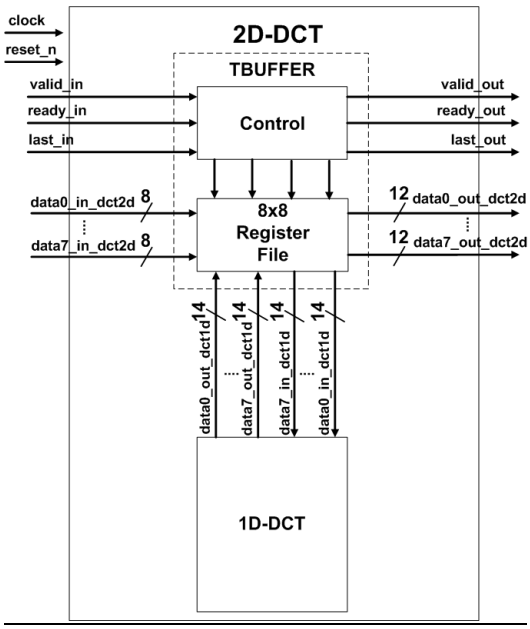

Figure 1: 2-D DCT Architecture

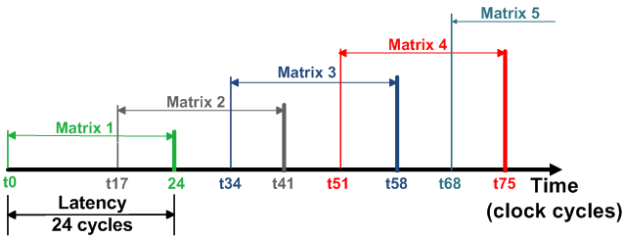

Figure 2: The 2-D DCT IP timing diagram

Figure 3 shows the 1-D DCT architecture. It has four butterflies that process eight 14-bit inputs. In a second processing level there are two butterflies and three adders. In a third level there are four rotates and a butterfly. A last level composed of four adders and eight shifters generates the eight 14-bit width outputs. As the 1-D DCT block is used for both row and column computations, a few control signals provided by the control unit are used to configure it.

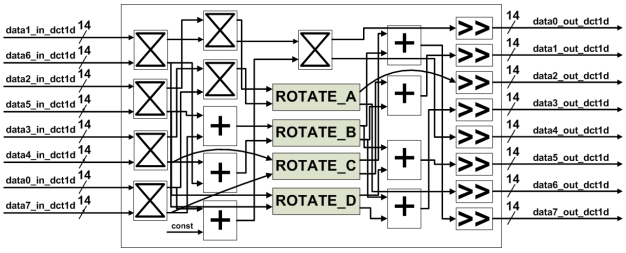

Figure 3: 1-D DCT Architecture

Figure 4 shows the rotate block. It consists of four adders, three shift-add multipliers and two shifters. Since the rotations are the main part of the algorithm and also responsible for its precision, no further modifications were made. Each rotate block needs a different set of constants which are used in the multiplications. The subsequent sum uses the same constant for all rotate blocks, just by alternating from the first to the second 1-D DCT computation. The shift amounts also alternate from the first to the second 1-D DCT computation, but are the same for all rotations.

Figure 5 shows the shift-add multiplier architecture. The input value is a 14-bit integer whereas the output uses an appropriate width so as to avoid any precision loss. The constant used in the multiplication of Figure 5 is 10498, thus corresponding to one of the multiplications needed in block ROTATE_D.

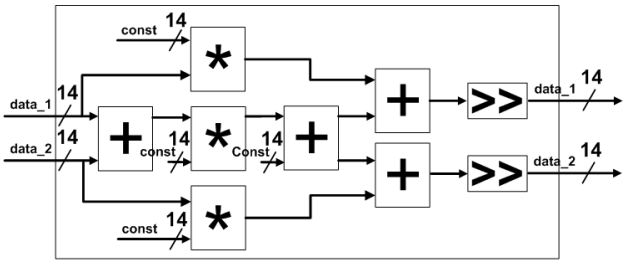

Figure 4: Rotate Block

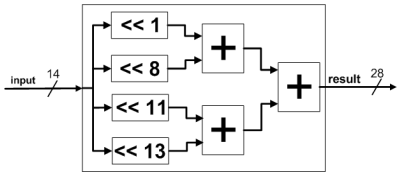

Figure 5: Multiplication as shift-add

4. SYNTHESIS RESULTS

The architecture was described in Verilog HDL and synthesized for the Xfab 350 nm CMOS standard cell technology using the Synopsys Design Compiler. To provide a fair comparison with Agostini's [9], we synthesized it for the same 350 nm technology. In both cases, the adder provided by Synopsys DC library was used (instead the carry lookahead adder used in Agostini's work [9]).

The synthesis results are shown in Table 3. Among the results, our Agostini's synthesis lead to the smallest estimated area (2.34 mm^2). The proposed architecture is the second smallest solution, being 37% larger than Agostini's. This difference comes from the parallelism used in the proposed architecture, which processes 8 elements at the same time (whereas Agostini's processes a single element). The proposed architecture area is 25% smaller than the Hsia's (Kinani does not report area).

Table 3 also reports total power consumption for the considered architectures. However, due to the different throughputs, it is not possible to establish a fair comparison in terms of power. In order to compare the efficiency, we used a power/performance efficiency metric (PPEM), given in mJ/Mpixels. Under such metric the proposed architecture shows an improvement of 5.47 times with respect to Agostini's. As compared to Kinani's, it shows to be 7.5 times more efficient. Nevertheless the overhead of Kinani's shape adaptive structure is at most 50% of the total DCT architecture. The proposed architecture is 2.4 times more efficient than Hsia's.

Considering that Agostini's architecture was synthesized with the same flow, the comparisons are more meaningful than the others. Besides, the large PPEM improvement of the proposed architecture, Synopsys DC reported a critical path slack that is 2.75 times longer than that reported for Agostini's architecture. Such result shows that the proposed architecture has a broader exploitable space for minimizing power consumption by applying low power techniques such as Multi-Vdd and Multi-Vt.

5. CONCLUSION

This paper presented an 8x8 2-D DCT architecture for high throughput low power IP core. High throughput is achieved by using an 8-element parallelism and a combinational 1-D DCT. The low power is achieved by avoiding the expensive pipeline structure, using a simple architecture built upon a 1-D DCT combinational block and a register file to implement the transposition. Such features lead to low clock frequency, resulting in lower switching activity.

The proposed architecture shows to be at least 2.4 times more efficient in terms of the considered mJ/Mpixels metric as compared to 2-D DCT architectures presented in related works. The reported critical path slack suggests a broader space to explore for further power improvements.

6. REFERENCES

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Transactions on Computers, vol. C-32, pp. 90-93, Jan. 1974.

[2] Home site of the JPEG and JBIG committees. Available from: <http://www.jpeg.org>.

[3] MPEG-2 video coder, ISO/IEC DIS 13818-2.

[4] J. D. Murray, W. Vanryper, Encyclopedia of Graphics File Formats. 2nd ed. Sebastopol: O'Reilly & Associates, 1996. 1116p.

[5] C. Loeffler, A. Ligtenberg and C. S. Moschytz, "Practical Fast 1D DCT Algorithm with Eleven Multiplications", Proc. ICASSP 1989, 988-991.

[6] Y. Arai, T. Agui, M. Nakajima, "A fast DCT-SQ Scheme for Images," Trans IEICE #71 (1988), 1095-1097.

[7] S. Hsia, C. Tsai, S. Wang and K. Hung, "Transposed-Memory Free Implementation for Cost-Effective 2D-DCT Processor". J. Signal Process. Syst. 58, 2 (Feb. 2010), 161-172.

[8] A. Kinane and N. O'connor, "Energy-efficient Hardware Accelerators for the SA-DCT and Its Inverse," J. VLSI Signal Process. Syst. 47, 2 (May. 2007), 127-152.

[9] L. Agostini, I. S. Silva, S. Bampi. "Multiplierless and Fully Pipelined JPEG Compression Soft IP Targeting FPGAs," Microprocessors and Microsystems, v. 31, p. 487-497, 2007.

[10] P. Massimino, DCT implementation site. Available from: <http://web.archive.org/web/20060707003114/http://skal.planet-d.net/coding/dct.html>

[11] IEEE Standard Specifications for the Implementations of 8x8 Inverse Discrete Cosine Transform. IEEE Std 1180-1990.

[12] Home site of Brazil IP project. Available from: <http://www.brazilip.org.br/>

[13] ARM. AMBA 3 Specification. 2009. Available from: <http://www.arm.com/products/solutions/AMBA3AXI.html>.

[14] M. Keating, D. Flynn, R. Aitken, A. Gibbons, K. Shi, *Low Power Methodology Manual for System-on-Chip Design*. Series: Series on Integrated Circuits and Systems, 2007.

TABLE III. SYNTHESIS RESULTS

Architecture	Process [nm]	Voltage [V]	P (Total) [mW]	mW/ (Mpixels/s)	Area [mm^2]	Latency [Clk cycles]	Speed [MHz]
Agostini*	350	3.3	41.67	2.19	2.34	163	19
Hsia [3]	350	3.3	15	0.96	4.00	22	60
Kinane [4]	350	3.3	15	3	n/a	142	n/a
Proposed	350	3.3	7.65	0.40	3.20	24	5

*. Our synthesis results.