Design of Energy-Efficient Discrete Cosine Transform using Pruned Arithmetic Circuits

Jeremy Schlachter, Vincent Camus, Christian Enz
Integrated Circuits Laboratory (ICLAB)
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
jeremy.schlachter@epfl.ch, vincent.camus@epfl.ch

***Abstract*—Inexact circuits and approximate computing have been gaining a lot of interest in order to improve performances and energy efficiency beyond the boundaries of conventional digital circuits. Image and video processing is one of the best candidate for applying such techniques. As one of the key building blocks, Discrete Cosine Transform (DCT) accelerators are investigated using pruned arithmetic circuits. A design methodology is presented in order to optimize both image quality and circuit performances. This work demonstrates that with such technique, savings are possible not only on arithmetic units, but in the entire accelerator hardware. Simulations show up to 12 % area and 10 % power savings with less than 20 dB PSNR degradation compared to the conventional DCT design.**

Keywords—Approximate computing, inexact circuits, pruning, DCT.

I. Introduction

In the past four decades, technology scaling has been the driving force behind the semiconductor industry. However, this trend is forecasted to end due to the rising cost and complexity of deep sub-micron technologies. *Approximate computing* has emerged as a major field of research as it could significantly improve energy efficiency and performances and rescue Moore's law [1]. Indeed, many applications are inherently tolerant to errors or approximations. Multimedia processing and applications involving human perception can easily accept small variations as long as they stay indiscernible by human senses. More surprisingly, weather forecasting algorithms that require a huge amount of calculations on high-performance computers have also been proven to be error tolerant [2]. Besides the error tolerance of applications, modern digital systems also suffer from over-engineering due to the lack of cross-layer considerations between application specifications and circuit design.

Many techniques have been proposed to exploit approximation at circuit level. The first attempt of trading accuracy versus energy consumption has been presented in the early 2000s with devices exploiting white noise to produce a probabilistic behavior [3]. However, this approach was unfruitful since noise levels in current technologies are much lower than predicted. An approach to potentially save energy is to exploit the quadratic relationship between supply voltage and power consumption with Voltage Over-Scaling [4], [5], which consists in reducing the voltage below the critical point where timing errors start to occur. Nevertheless, timing failures are difficult to predict and generally in an abrupt loss of functionality dramatically above a critical threshold. Also compensation circuits are complex and induce undesired circuit and system level overhead.

An alternative approach is to exploit the graceful and deterministic application specific quality metric degradation offered by approximate arithmetic operators. Different techniques such as logic minimization [6], simplified full adder cells [7] or speculative circuits [8] have recently been proposed. As a proof of concept, the first inexact adders (where full adder cells have been pruned from various adder architectures) have been fabricated [9] and measurements have demonstrated up to one order of magnitude savings. However, most of those techniques are based on manual designs or tweaks, and have not yet been integrated in the standard digital flow.

Gate-Level Pruning (GLP) [10] tries to address this issue by automatically generating approximate arithmetic circuits starting from a conventional design and using existing standard digital design tools and flow. However, these components are generally part of a complex system and occupy only a small portion of the entire chip area. It is consequently not clear how the GLP technique or other inexact arithmetic circuits can be exploited to trade off accuracy for area and energy in a system built out of multiple arithmetic circuits but also memories and other IP blocks, and where errors produced by a single adder could either be amplified or attenuated in the following stages of the computation.

This paper illustrates how the GLP technique can be exploited in conjunction with a design space exploration to build a hardware accelerator: the Discrete Cosine Transform (DCT), which is one of the most computationally intensive element for many image and video processing compression algorithms such as JPEG or MPEG. The demonstration is made that approximating arithmetic operators not only reduce the power consumption of adders and multipliers, but also improves the energy-efficiency of the overall system they are placed in by reducing the switching activity and by lowering the required amount of memory and registers. This work does not claim to present a novel type of DCT, but it demonstrates how energy-quality tradeoffs can be achieved by applying inexact design techniques on existing state-of-the-art architectures.

The remainder of this paper is organized as follow: section II details how to automatically generate approximate arithmetic circuits with varying error levels using the GLP technique, section III explains the conventional DCT architecture taken as a reference design. Finally, section IV describes the methodology used to apply the GLP at accelerator level and shows the resulting energy accuracy tradeoff for the entire DCT.

This work was supported by the NanoTera "IcySoC" project and the Swiss National Science Foundation grant No 200021-144418.

II. Inexact Arithmetic Circuit Design using Gate-Level Pruning

Gate-Level Pruning is a CAD technique to automatically generate inexact circuits starting from a conventional design by adding only one small step in the digital design flow. The CAD framework is presented in Fig. 1.

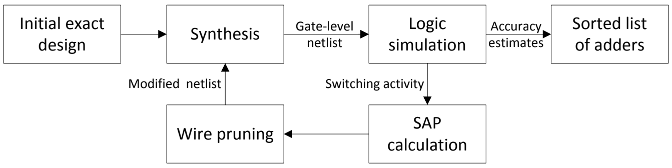


Fig. 1: CAD framework for Gate-Level Pruning.

Any exact circuit can be represented by a directed acyclic graph as depicted in Fig. 2, where the nodes are components such as gates, and whose edges are wires. The decision to prune a node is based on two criteria: the significance, which is a structural parameter, and the activity or toggle count. The nodes with the lowest Significance-Activity Product (SAP) are pruned first. By doing so, the error magnitude grows with the amount of pruning. Alternatively, depending on the application's requirement, the designer may choose to prune nodes according to the activity only, in order to minimize the error rate.

The activity of each wire is extracted from the SAIF file (Switching Activity Interchange Format) obtained through gate-level hardware simulations. This file contains the toggle count of each wire, as well as the time spent at the logic levels 0 and 1 respectively. In order to get an accurate activity estimation, the system should be simulated with an input stimulus representative of the *real operation* of the circuit. The more the simulation is realistic, the more the toggle count is accurate and leads to an efficient pruning.

The significance of each primary output is set by the designer depending on the application's requirement. In this paper, pruning is applied on several arithmetic circuits where each primary output is weighted by a power of two. It is therefore worth applying a weighted significance attribution, where each bit position has a significance two times higher than the previous when moving from the Least Significant Bit (LSB) to the Most Significant Bit (MSB). Reverse topological graph traversal is then performed to compute each nodes' significances as follows:

$$\sigma_i = \sum \sigma_{desc(i)} \tag{1}$$

where σ_i is the significance of the node i and $\sigma_{desc(i)}$ is the significance of the direct descendants of node i. An example of weighted significance attribution is shown in Fig. 2.

Once the significance and activity is determined, the nodes, i.e. gates and their corresponding wires, are ranked according to their SAP. The ones with the lowest SAP are disconnected from the verilog netlist, and an incremental re-synthesis is performed in order to remove or replace the unconnected gates.

This methodology can easily be applied to any arithmetic circuit to trade a little accuracy for significant area and power savings. However it is unclear how the savings achieved on an single operator can be translated at system level, where memories, registers and other IP blocks can dominate silicon area and power consumption. The DCT is taken as an example to validate the use of Gate-Level Pruning in a complex system.

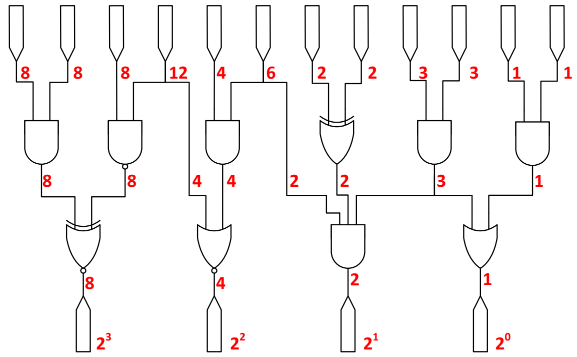


Fig. 2: Directed acyclic graph representation of a gate level netlist and the associated significance attribution

III. The Discrete Cosine Transform Accelerator

A. Conventionnal DCT

DCT algorithms and architectures have been extensively studied. Image encoding algorithms used for instance in JPEG encoding generally compute the DCT per pixel blocks. The following work considers the example of 8x8 pixel blocks DCT, but could be extended to other block sizes and architectures. Efficient implementations are generally based on distributed arithmetic computations [11], and is taken as starting point for the following example.

A 2D DCT used in image encoding can be split in two single stage DCTs interleaved with transpose memory as shown in Fig. 3. The 8-point 1D-DCT w_k of a data sequence x_i is defined by:

$$w_k = \frac{a_k}{2} \sum_{i=0}^{7} x_i \cos\left[\frac{(2i+1)k\pi}{16}\right] \tag{2}$$

$$\text{with } a_k = \begin{cases} 1/2, & k = 0 \\ 1, & k = 1...7 \end{cases}$$

This can also be expressed in its matrix form:

$$W = T \cdot X \tag{3}$$

where T is an 8 x 8 matrix in the case of an 8 point DCT and X and W are row and column vectors. Using the symmetry property of T, (3) can be decomposed as follow for even / odd 1D DCT calculations:

$$\begin{bmatrix} w_0 \\ w_2 \\ w_4 \\ w_6 \end{bmatrix} = \begin{bmatrix} c_4 & c_4 & c_4 & c_4 \\ c_2 & c_6 & -c_6 & -c_2 \\ c_4 & -c_4 & -c_4 & c_4 \\ c_6 & -c_2 & c_2 & -c_6 \end{bmatrix} \begin{bmatrix} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} w_1 \\ w_3 \\ w_5 \\ w_7 \end{bmatrix} = \begin{bmatrix} c_1 & c_3 & c_5 & c_7 \\ c_3 & -c_7 & -c_1 & -c_5 \\ c_5 & -c_1 & -c_7 & c_3 \\ c_7 & -c_5 & c_3 & -c_1 \end{bmatrix} \begin{bmatrix} x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{bmatrix} \tag{5}$$

where $c_k = \cos(\frac{k\pi}{16})$. It can be seen from (2) that the DCT is computationally intensive, and requires a large amount

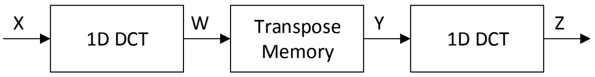


Fig. 3: 2D DCT architecture based on 1D stages

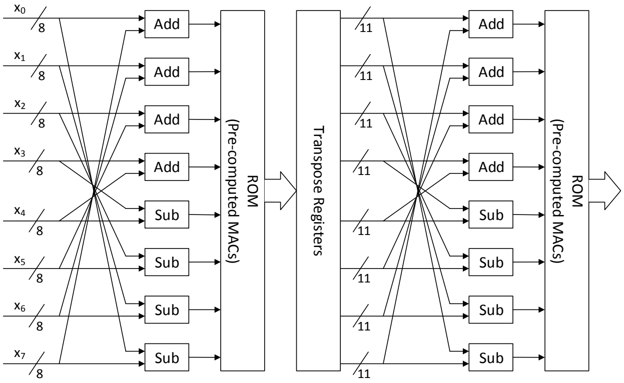


Fig. 4: Architecture of the 8 x 8 2D DCT.

of multiplications which are power hungry. Plenty of DCT architectures have been proposed in the literature. However, since the scope of the paper is to improve energy-efficiency, a low power multiplier-less DCT architecture based on row-column parallel distributed arithmetic has been chosen. Fig. 4 shows this implementation of the 8 x 8 2D DCT where only 4 adders and 4 subtractors are required to compute the right part of (4) and (5). The final 1D DCT is obtained by looking-up pre-computed multiply and accumulate (MAC) coefficients stored in a Read-Only Memory (ROM).

B. Quality testing

Fig. 5 sketches the test setup used to characterize the DCT for image processing. First, the DCT of an image sample is computed with the hardware under test. Image is then reconstructed using a behavioral inverse transform, i.e. with infinite precision. The quality of the reconstructed image compared to the original image is evaluated by calculating the Peak Signal-to-Noise Ratio (PSNR) between the two images as follow:

$$\text{PSNR} = 10 \log_{10} \left(\frac{D^2}{\text{MSE}} \right) \tag{6}$$

where MSE is the mean squared error between the original and the reconstructed image and D is the maximum possible pixel value, here 255, considering 8-bit pixel representation. With a sample *Lena* picture transformed by the conventional 2D DCT shown in Fig. 4, the PSNR is equal to 48 dB. Image quality is limited mainly due to the use of fixed point arithmetic. As conventional designs are already lossy, it can be acceptable to trade some more accuracy in exchange for power and silicon area savings.

IV. Approximate DCT Design

A. Design methodology

The 2D DCT described in III-A has been synthesized with an industrial 65 nm technology at a clock frequency of 1.25 GHz.

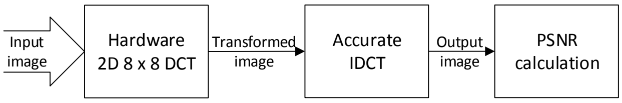


Fig. 5: Test setup for quality measurement

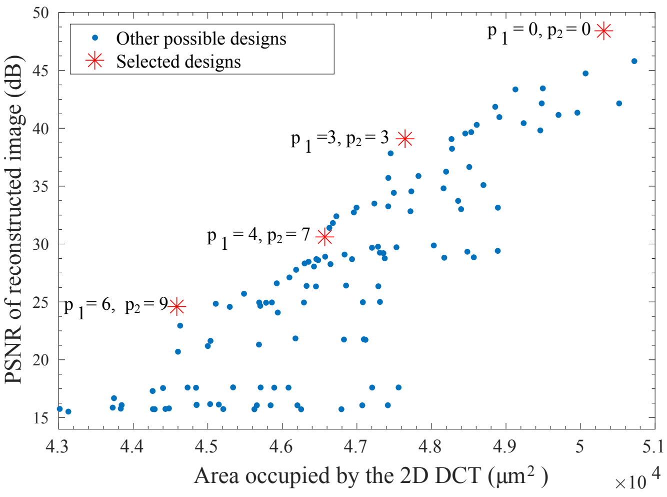


Fig. 6: Image quality versus circuit area

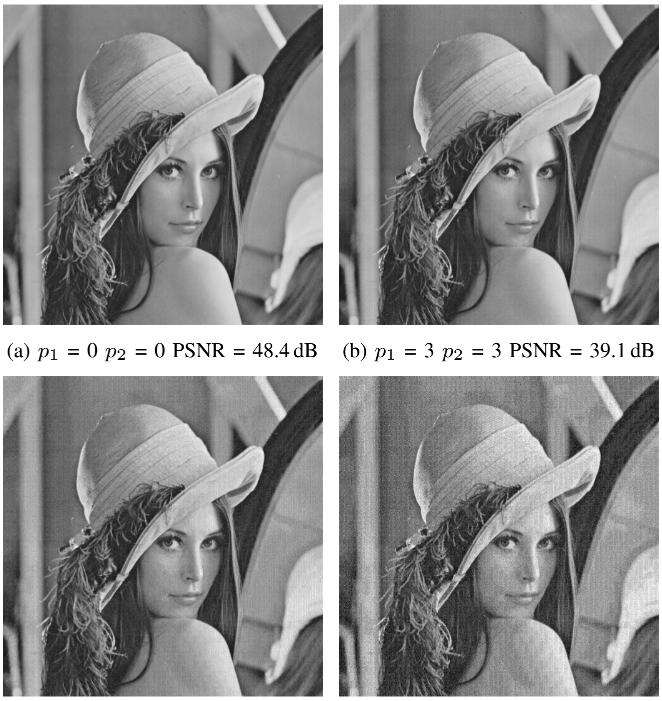
The resulting circuit is used as a the reference to apply the Gate-Level pruning to each of the 16 adders and subtractors. Seeing that each of these components have slightly different architectures due to differences in timing paths, and considering that the switching activity differs from one to another, pruning is applied individually on each of the 16 operators. Besides, each can have a different impact on the final error bound. It is consequently required to explore the design space to find out the best possible combination of inexact adders in order to minimize the quality loss and maximize the savings. The synthesized adders and subtractors are built out of 45 standard cells in average. It is therefore worth pruning up to 10 nodes for fine-tuning the accuracy. Higher pruning would dramatically degrade the image quality. For 10 levels of pruning considered per adder and subtractor (the exact operator plus 10 pruned ones), there are 11^{16} possible design combinations. For practical reasons such as computing resources, it is clearly not possible to run 11^{16} synthesis and hardware simulations to find out the optimal design.

A good solution to narrow the design space is to apply the same level of pruning p_i to each adder and subtractor inside a given stage i. As the bit-width is the same within a stage, the degradation of arithmetic accuracy is progressive. With this approach, there are $11^2 = 121$ possible combinations left.

Synthesis shows that the area occupied by the 16 adders and subtractors depicted in Fig. 4 represent a small part of the entire conventional 2D DCT area. Hence, a simple swap between exact and approximate operators would lead to very limited savings. Nevertheless, re-synthesizing the full design with pruned operators can eliminate unused ROM and unnecessary registers thanks to logic simplification and constant propagation implemented in the synthesis tool. This results in attractive power and area savings.

B. Results

Fig. 6 shows the image quality versus area savings for the implemented DCTs. Each point corresponds to a combination (p_1, p_2) in $[0, 10]^2$. This figure highlights the broad diversity of design options offered using this methodology. For a given



(a) p_1 = 0 p_2 = 0 PSNR = 48.4 dB (b) p_1 = 3 p_2 = 3 PSNR = 39.1 dB

(c) p_1 = 4 p_2 = 7 PSNR = 30.6 dB (d) p_1 = 6 p_2 = 9 PSNR = 24.6 dB

Fig. 7: Pictures of Lena resulting from the test setup using the conventional DCT (a) and the approximate versions (b,c,d). p_i denotes the number of pruned nodes per adder and subtractor in stage i.

image quality requirement, pruning of operators in such a complex system allows to precisely match design specifications with an optimal circuit efficiency.

Keeping in mind that the goal of approximate circuits is to trade a little accuracy for the maximum area and power savings, only designs along the upper envelope of the plot in Fig. 6 are of interest since they maximize the gains with minimum quality loss.

Fig. 7 shows reconstructed *Lena* pictures obtained from four selected DCT implementations (the red stars in Fig. 6 highlight those designs). Conventional DCT has been used for Fig. 7a, while the three others have been obtained using three pruned designs representative of possible the area-accuracy tradeoff plotted in Fig. 6. On the one hand, it is possible to save up to 12 % area at the cost of almost imperceptible errors. On the other hand, for designs achieving the highest area reductions, artefacts start to appear on the edges of the 8x8 pixel blocks.

For the selected designs, power consumption is estimated based on gate-level simulations monitoring switching activity of the *Lena* picture processing. Results are summarized in Table I. Despite adders and substractors represent only 4 % of the overall DCT area, re-synthesis of the design with pruned operators enables larger savings over the entire system, as explained in IV-A. For the case ($p_1 = 6$, $p_2 = 9$), 10 % savings are achieved for both area and power.

V. Conclusion

Research in approximate circuit has mainly focused on arithmetic units, which are key building blocks of digital systems but generally represent a small fraction of the overall circuit. This work has presented a methodology to design DCT accelerators using gate-level pruning, taking benefit of its systematic approach and good integration in the digital flow. Several combinations of pruned adders and subtractors have been investigated. Despite arithmetic circuits occupy less than 4 % of the total DCT area, the re-synthesis of the entire DCT with pruned operators enables up to 12 % area and 10 % power savings over the entire system.

TABLE I: Power, area and quality of the 4 selected DCTs

Pruning level	PSNR (dB)	Normalized Power	Normalized area
p_1 = 0 p_2 = 0	48.4	1	1
p_1 = 3 p_2 = 3	39.1	0.96	0.94
p_1 = 4 p_2 = 7	30.6	0.94	0.92
p_1 = 6 p_2 = 9	24.6	0.90	0.88

References

[1] K. Palem, A. Lingamneni, C. Enz, and C. Piguet, "Why design reliable chips when faulty ones are even better," in *ESSCIRC (ESSCIRC), 2013 Proceedings of the*, Sept 2013, pp. 255–258.

[2] P. Düben, J. Schlachter, Parishkrati, S. Yenugula, J. Augustine, C. Enz, K. Palem, and T. N. Palmer, "Opportunities for energy efficient computing: A study of inexact general purpose processors for high-performance and big-data applications," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, ser. DATE '15. San Jose, CA, USA: EDA Consortium, 2015, pp. 764–769. [Online]. Available: http://dl.acm.org/citation.cfm?id=2755753.2755927

[3] K. V. Palem, "Energy aware computing through probabilistic switching: A study of limits," vol. 54, no. 9. IEEE, 2005, pp. 1123–1137.

[4] S. Ghosh, S. Bhunia, and K. Roy, "Crista: A new paradigm for low-power, variation-tolerant, and adaptive circuit synthesis using critical path isolation," vol. 26, no. 11. IEEE, 2007, pp. 1947–1956.

[5] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner *et al.*, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on*. IEEE, 2003, pp. 7–18.

[6] A. Lingamneni, C. Enz, K. Palem, and C. Piguet, "Parsimonious circuits for error-tolerant applications through probabilistic logic minimization," in *Integrated Circuit and System Design. Power and Timing Modeling, Optimization, and Simulation*. Springer, 2011, pp. 204–213.

[7] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," vol. 32, no. 1, Jan 2013, pp. 124–137.

[8] V. Camus, J. Schlachter, and C. Enz, "Energy-efficient inexact speculative adder with high performance and accuracy control," in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 45–48.

[9] A. Lingamneni, K. K. Muntimadugu, C. Enz, R. M. Karp, K. V. Palem, and C. Piguet, "Algorithmic methodologies for ultra-efficient inexact architectures for sustaining technology scaling," in *Proceedings of the 9th Conference on Computing Frontiers*, ser. CF '12. New York, NY, USA: ACM, 2012, pp. 3–12. [Online]. Available: http://doi.acm.org/10.1145/2212908.2212912

[10] J. Schlachter, V. Camus, C. Enz, and K. Palem, "Automatic generation of inexact digital circuits by gate-level pruning," in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, May 2015, pp. 173–176.

[11] S. Yu and J. Swartziander, E.E., "Dct implementation with distributed arithmetic," vol. 50, no. 9, Sep 2001, pp. 985–991.