A Novel Pruned-Based Algorithm for Energy-Efficient SATD Operation in the HEVC Coding

Leonardo Bandeira Soares[1], Cláudio Machado Diniz[2], Eduardo Antonio César da Costa[2], Sergio Bampi[1]
[1]Graduate Program on Microelectronics (PGMicro)
Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil
[2]Graduate Program on Electronic Engineering and Computing (PPGEEC)
Catholic University of Pelotas (UCPEL), Pelotas, Brazil
{lbsoares, bampi}@inf.ufrgs.br; {claudio.diniz, eduardo.costa}@ucpel.edu.br

Abstract— **This paper proposes a novel approximate computing algorithm for the Sum of Absolute Transformed Differences (SATD) to meet energy efficiency in CMOS accelerator circuits. It is based on the pruning of least significant coefficients in the 2-D Hadamard Transform (HT) which is the most compute intensive kernel in the SATD. The SATD is a metric for block matching that is used in video coding standards like the new High Efficiency Video Coding (HEVC). This metric is used to provide better results in mode decision when compared to the Sum of Absolute Differences (SAD) at the expense of larger amount of arithmetic operations as well as higher energy consumption. We present 6 different approximate SATD 4x4 architectures that were synthesized for a 45 nm PDK. Results for the approximate architecture with 10 discarded HT coefficients show energy per operation reduction of 70.7% and BD-PSNR reduction of just -0.008 dB, for a 1080p video sequence.**

Keywords—Approximate Computing; CMOS Energy Efficiency; Video Coding; Low Power CMOS design.

I. INTRODUCTION

The semiconductor industry faces challenges at each new CMOS technology node. One of them is the power density increase, which is related to the physical limitations to scale the nominal supply voltage by the same factor used to scale down the device channel length. The main consequence is that the power per silicon area has been substantially increased, so that operational blocks of a chip need to be turned off frequently to allow for heat dissipation [1]. Another CMOS power concern is associated with portable and battery-powered devices which demand intensive computation and power-hungry applications. Nowadays, it is mandatory to shift from performance-driven to energy-efficient oriented CMOS digital circuit design techniques. Energy efficiency is defined in [2] as being the maximum number of operations per energy budget or the minimum consumed energy per unit operation (*i.e.* instruction fetching/decoding, addition, etc).

Among the vast and different proposed energy-efficient techniques, the approximate computing emerged to increase performance and to reduce power dissipation [3]. The key approach in approximate hardware is to reduce the accuracy in the computation by designing simpler circuits to speed up the critical path timing, and to consume less power. Approximate computing techniques take advantage of error-tolerant applications which do not need high accuracy all the time but only good enough results for output perceptual quality. For example, multimedia applications (e.g. video coding, audio filtering, image processing, and so on), highly demanded by current portable devices, are intrinsically related with human senses. Since human sensors process analog information and have difficulty to realize digital approximations [4], then multimedia signals are in fact error-tolerant applications. In other words, it is possible to adopt approximate computing techniques to improve energy efficiency in multimedia applications by properly exploiting the user experience at different levels of quality.

Video coding is one application which demands high computational effort to provide high video compression ratio. HEVC is the new video coding standard [5], which improves compression by 50% compared to the previous H.264/AVC standard [6], while increasing computational effort by up to 3.2x [7]. Hence, energy-efficient techniques for HEVC are mandatory to accomplish the severe power/performance requirements. In terms of block matching for Fractional Motion Estimation (FME), the HEVC adopts the use of SATD. According to [7], for random access configuration in the HEVC Test Model (HM) reference software [8], FME and SATD represent 55% and 16% of the computational effort demanded by the entire encoding process, respectively.

Since the SATD is a compute intensive kernel in HEVC, we can leverage the condition of good enough block matching in order to improve the energy-efficiency. In [9] the use of approximate adders is proposed in the less complex and less accurate SAD metric for block matching technique used in MPEG encoders. In that work, maximum power savings of 42% for very low video resolutions (CIF resolution) are shown. Our approximate algorithm copes with new challenges imposed by the more complex HEVC standard and the use of higher video resolutions (e.g. 1080p and 1600p). To the best of our knowledge, there is no previous work that addresses approximate computing techniques for SATD. In our work the key point is to perform approximation in the 2-D HT, since this is the most compute intensive kernel in the SATD. Our approximate algorithm is driven by pruning the least significant coefficients from the 2-D HT. We show 6 different approximate architectures for the case study on SATD 4x4. The approximate architecture which discards 10 coefficients presents 70.7% of energy per operation reduction while the BD-PSNR decreases 0.008 dB in comparison with the precise SATD 4x4 for a 1080p video sequence.

The remainder of this paper is organized as follows: Section II presents a basic overview in SATD metric for HEVC coding. Our approximate algorithm is presented in Section III followed by the case study on SATD 4x4 results in Section IV. Finally, the conclusions are drawn in Section V.

II. Sum of Absolute Transformed Differences Overview

The SATD is a metric that measures distortion between two blocks in video coding. This distortion is evaluated by block matching methods for motion estimation. For the SATD computation the differences between two blocks are transformed by the 2-D forward HT as shown in (1).

$$W = H \cdot Y \cdot H^T \tag{1}$$

In (1), W and Y denote the transformed coefficients and the input matrix of differences, respectively. The H and H^T refer to the transform matrix and its transposed form, respectively. For instance, the H can be seen in (2) considering a block size of 4x4 pixels.

$$H = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{2}$$

Based on that, the SATD is calculated with the aforementioned HT as shown in (3), where $w_{i,j}$ denotes the HT coefficient from the i^th^ row and j^th^ column.

$$SATD = \sum_{i,j} |w_{ij}| \tag{3}$$

III. Our Coefficient Pruning-based Approximation Algorithm

In the following subsections we show the computational cost of the baseline SATD architecture. Then we detail next our coefficient pruning-based algorithm to perform approximation in the HT, and its resulting approximate architectures with an analysis of computational cost reduction.

A. Computational Cost of the Baseline SATD Architecture

In this work we assume as baseline a parallel and precise SATD hardware architecture. For example, the parallel precise SATD 4x4 architecture is shown in Fig. 1. The input x_{ij} denotes the difference between two blocks in the i^th^ row and j^th^ column. The first step in the architecture is the horizontal transform. The transformed samples y_{ij} are registered and are delivered to the vertical transform. Then, the absolute operator is applied in the HT coefficients w_{ij} followed by the adder tree sum. One can observe that these last two steps of absolute operators and sum implement the SAD 4x4 architecture.

Both the horizontal and vertical transforms (*i.e.* the first two steps in Fig. 1) are implemented by the internal structure presented in Fig. 2. One can observe that inside the vertical transform we use the intermediate term v_{ij} that is important for our coefficient pruning-based algorithm as explained further on Section II-B. For the horizontal transform we define the intermediate term as being h_{ij}. The horizontal and vertical transforms refer to the HT presented in (1) and represent an additional computational cost when compared to the SAD computation. Based on that, Table I summarizes the computational cost in terms of operators (*i.e.* adders and absolute operator) for the SATD and SAD with three different block sizes of 2x2, 4x4, and 8x8, while considering architectures of 2-D Hadamard Transform as shown in Fig. 1.

TABLE I. SATD and SAD Computational costs

	SATD			SAD		
block size	2x2	4x4	8x8	2x2	4x4	8x8
adders	11	79	447	3	15	63
absolute operators	4	16	64	4	16	64
total	15	95	511	7	31	127

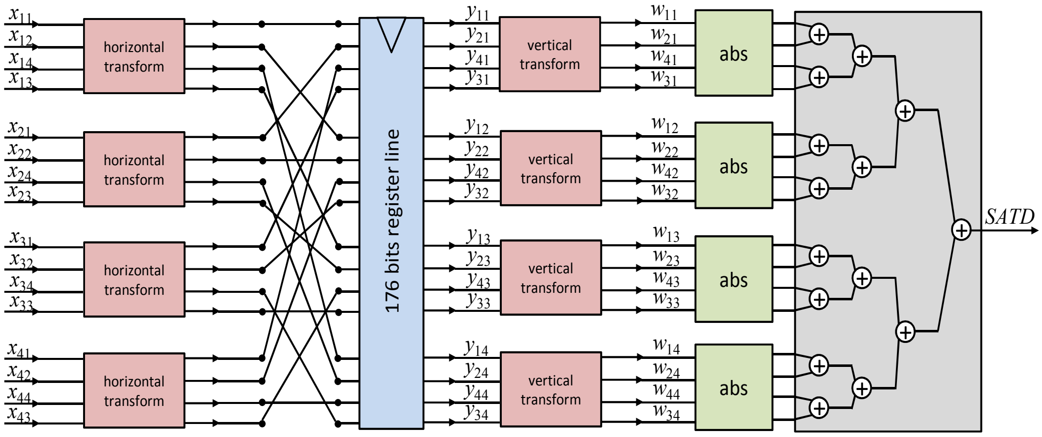

Fig. 1. Precise SATD 4x4 hardware accelerator parallel architecture.

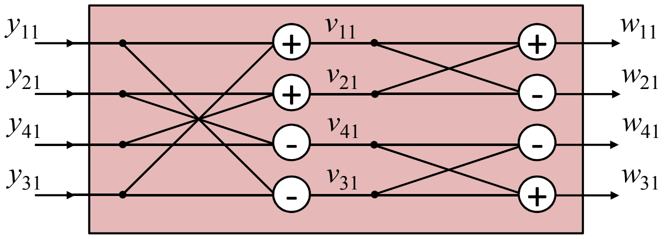

Fig. 2. Internal structure of horizontal and vertical transforms.

Table I clearly shows the additional hardware cost of 3.1X when performing fully parallel the SATD 4x4 instead of the SAD 4x4. This cost is 4X higher when considered the SATD 8x8 in comparison with the SAD 8x8. Based on that, this is our motivation to propose the pruning-based algorithm in order to approximate the HT by reducing its hardware cost in terms of adders count, as well as power consumption.

B. The Proposed Pruning-based Algorithm

The proposed pruning-based algorithm shown in Algorithm 1 is independent from the block size of the SATD. Our algorithm uses two data structures: i) a stack S containing the HT absolute coefficients w_{ij} organized from the top to the bottom with the least to the most significant coefficients, ii) a tree R of dependencies among the prior terms used in the HT to compute each coefficient w_{ij}. The stack is important to decide the order of the coefficients being discarded from the matrix W shown in (1). The tree is used by the proposed algorithm to solve dependencies among the discarded coefficients w_{ij} and prior terms that can be pruned from the architecture because they are not used by the remaining HT coefficients. The proposed algorithm also takes as input the number of coefficients n to be discarded. The output is the approximate SATD RTL VHDL netlist generated according to the remaining terms in the pruned tree R.

Before explaining the algorithm, an example of the tree of dependencies is shown in Fig. 3, since this is the central structure for the algorithm. This example refers to the sub-tree related to the vertical transform presented in Fig. 2. All the coefficients w_{ij} are leaves in the tree. They are computed by the prior terms denoted as father nodes. For instance, the prior terms v_{31} and v_{41} are used only for the computation of the w_{31} and w_{41} coefficients. Hence, if those coefficients are discarded, the proposed algorithm must prune from the tree v_{31}, v_{41}, and $y_{11} - y_{31} + y_{41} - y_{21}$ terms. It is worth to mention that all the remaining terms from the tree must be preserved, since the coefficients w_{11} and w_{21} were not discarded.

The proposed algorithm has an external loop that is performed n times to remove at each iteration the next least significant coefficient and its exclusive associated prior terms. The n parameter can range from 1 up to 16 (*i.e.* no use of HT). The *pruning* Boolean variable is used to control the pruning iteration in line 5. Each coefficient is extracted from the top of the stack S with the *pop* function. Then, the extracted coefficient is searched in the leaves of the tree R. Once the coefficient is found, then its location is stored in the variable l and the pruning process starts. The next step is to visit its father node and store the location in variable k. Since the location of the coefficient to be discarded is stored in variable l, then we use this information to effectively remove this coefficient from the tree. The next step is to evaluate if the current father node has a remaining child. If this verification is true, the pruning must stop, since there are coefficients or terms that still depend on the current father node. On the other hand, the current father node can also be removed. Before removing the current father node, it is important to verify if this node is also the root node.

Algorithm 1 The pruning-based algorithm for SATD

Input: S – a stack of coefficients sorted by significance
R – a tree of dependencies among HT terms
n – the number of coefficients to be discarded
Output: Q – the approximate SATD RTL netlist

```
1: while $n > 0$ do
2:     $pruning$ := TRUE
3:     $s := pop(S)$
4:     $l := search\_leaf(s,R)$
5:     while $pruning$ do
6:         $k := father(l,R)$
7:         $remove(l,R)$
8:         if $has\_child(k,R)$ then
9:             $pruning$ := FALSE
10:        else
11:            if $is\_root(k,R)$ then
12:                $remove(k,R)$
13:                $pruning$ := FALSE
14:            else
15:                $l := father(k,R)$
16:                $remove(k,R)$
17:                if $has\_child(l,R)$ then
18:                    $pruning$ := FALSE
19:                end if
20:            end if
21:        end if
22:    end while
23:    $n := n - 1$
24: end while
25: $Q := generate\_netlist(R)$
```

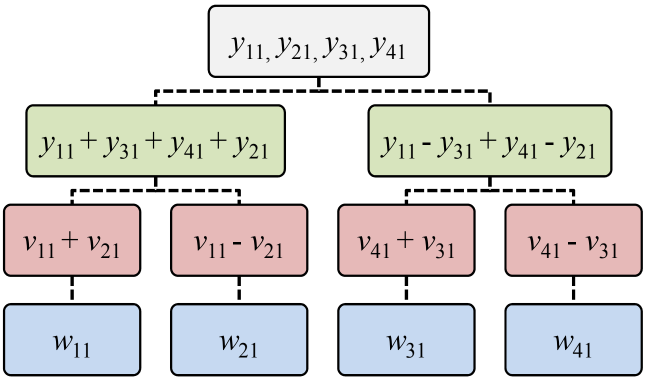
$y_{11}, y_{21}, y_{31}, y_{41}$
$y_{11} + y_{31} + y_{41} + y_{21}$
$y_{11} - y_{31} + y_{41} - y_{21}$
$v_{11} + v_{21}$
$v_{11} - v_{21}$
$v_{41} + v_{31}$
$v_{41} - v_{31}$
w_{11}
w_{21}
w_{31}
w_{41}

Fig. 3. Example of a tree of dependencies for the vertical transform.

If true, then this node is removed and the pruning must be stopped. This is because there are no more coefficients or terms to be pruned. Otherwise, is important to store the location of the grandfather in *l* before removing the current father. The next step is to verify if the grandfather node has not remaining child. When false, the pruning iteration (*i.e.* line 5) must continue to remove the grandfather node and possibly more prior terms. On the other hand, this pruning has to stop. This is because the grandfather still has child and cannot be pruned from the tree.

Until now, this paper has not discussed how the significance can be measured among the HT coefficients. In this work we define significance as being the average magnitude of each Hadamard transformed coefficient considering a set of video sequences from different classes. Once the stack of most significant coefficients is determined, then we perform the proposed algorithm to prune the SATD architecture based on prior average information. For example, in Fig. 4 we show the average magnitude for each coefficient in 4x4 HT considering 4 video sequences (*i.e.* from class A, B, C and D, as presented in Table II).

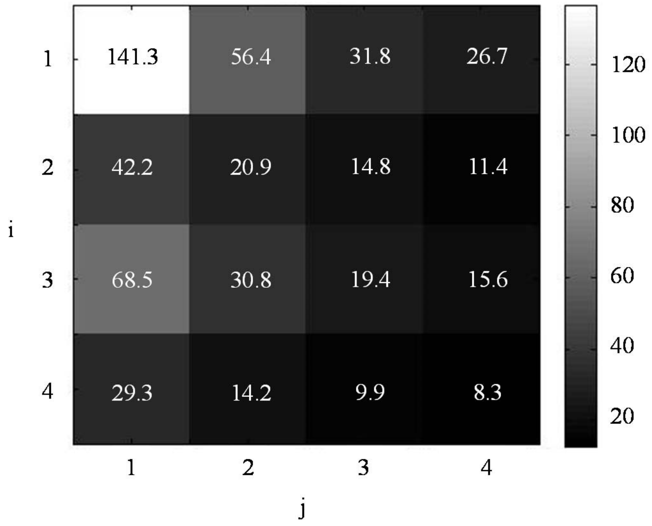

Fig. 4. Average magnitude of each coefficient in 4x4 HT (4 videos).

For instance, if we want to discard the ten least significant coefficients from the precise example in Fig.1, then the following coefficients will be removed: w_{44}, w_{43}, w_{24}, w_{42}, w_{23}, w_{34}, w_{33}, w_{22}, w_{14}, and w_{41}. One can observe that, based on our proposed algorithm, for each discarded coefficient the tree of dependencies is pruned to reduce the number of cells in the circuit. The resulting approximate SATD architecture can be verified in Fig. 5, where the approximate architecture does not have a complete horizontal or vertical transform. The pruned horizontal and vertical transform blocks have fewer adders than the complete ones. The reduction in the number of adders for the approximate SATD architecture in comparison with the precise one is 37.9%. In conclusion, the additional cost of the approximate SATD architecture shown in Fig. 5 is 1.7X when compared to SAD 4x4 architecture. In Table I, we show that the precise SATD 4x4 has an additional cost of 3.1X which has 1.8X more cost than the approximate SATD architecture with 10 least significant coefficients being discarded.

IV. Results

In this section we show results for the case study on SATD with block size of 4x4. First, we present the experimental setup followed by the results in terms of video quality, compression and energy efficiency.

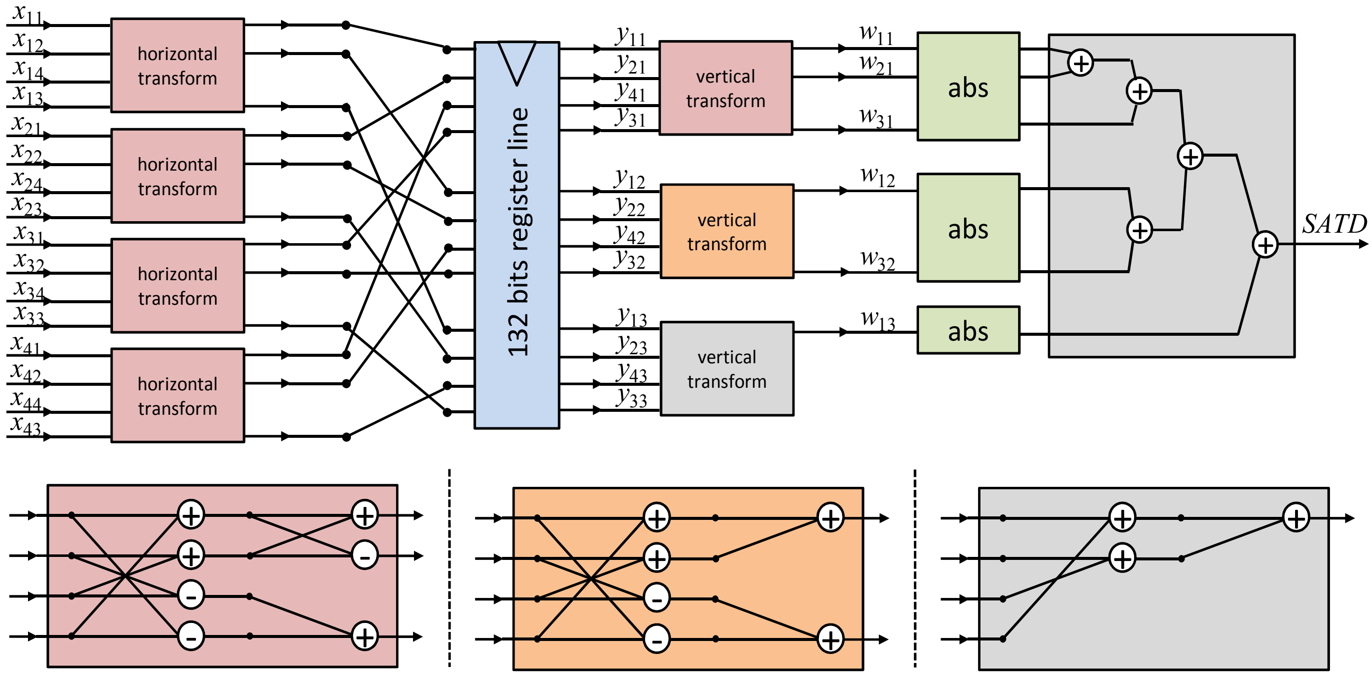

Fig. 5. Approximate SATD architecture with 10 discarded coefficients.

A. *Experimental Setup*

In order to analyze the video quality and compression, we use the HM 16.0 [8]. In the HM scope we encode 64 frames of 4 videos from the common test conditions [10] as shown in Table II.

TABLE II. VIDEO SEQUENCES SPECIFICATION

Class	Video sequence name	Resolution
A	Traffic	2560x1600
B	BQTerrace	1920x1080
C	RaceHorses	832x480
D	BlowingBubbles	416x240

We have selected the random access configuration (instead of all intra or low delay) because it is the most complex and most generic configuration for picture ordering. In order to evaluate the quality loss and the bit-rate increase, we run each video sequence for the following quantization parameters: 22, 27, 32, and 37. This is because we integrate the resulting PSNR and bit-rate points to evaluate the Bjøntegaard-Delta bit-rate (BD-BR) and PSNR (BD-PSNR). For BD-BR and BD-PSNR metrics we use the precise SATD 4x4 results as reference.

The encoding process is performed for the precise SATD 4x4 and for the range of 1 to 16 discarded coefficients. Since we consider the average significance to prune the SATD operation, the discarded coefficients for all the video sequences are performed in the same order. For the approximation where the 16 coefficients are discarded we define as being the SAD 4x4.

For the synthesis results we perform a multiple timing synthesis using the Cadence RTL Compiler™ based on the bisection method to find the maximum frequency of operation for each design under evaluation. All the circuits are mapped to a 45nm Nangate FreePDK [11]. For the iso-performance analysis we selected the maximum frequency achieved by the precise SATD 4x4 architecture since this design present the lowest maximum achievable frequency when compared to the other approximate designs. The power results were obtained through switching activity extraction from 10,000 real 4x4 blocks with residues from 4 video sequences.

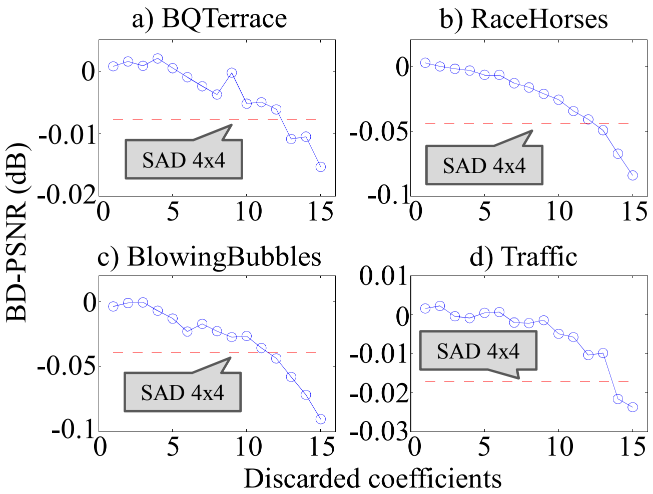

Fig. 6. BD-PSNR results for approximate SATD architectures.

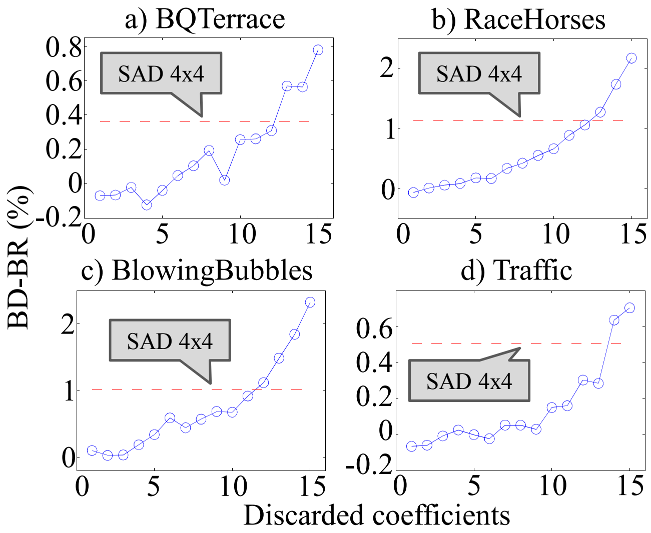

Fig. 7. BD-BR results for approximate SATD architectures.

B. *Video Quality and Compression Results*

Video quality (BD-PSNR) and video compression (BD-BR) results can be seen in Fig. 6 and Fig. 7, respectively. In the graphs we show discrete points for each configuration of discarded coefficients and the dotted red line represents the SAD 4x4 (*i.e.* when 16 coefficients are discarded in the 4x4 HT). In fact, this line represents a boundary to evaluate if there are configurations ranging from 1 to 15 discarded coefficients that has worse results than using the SAD 4x4. Based on that, for all the cases the SAD 4x4 presents better results than some pruned configurations. For instance, for the BQTerrace sequence both BD-PSNR and BD-BR SAD 4x4 results has lower degradation and higher compression than configurations from 13 up to 15 discarded HT coefficients. Therefore, is preferable to use the SAD 4x4 instead of those configurations because SAD 4x4 will be a more energy-efficient solution with better results regarding BD-BR and BD-PSNR.

In terms of BD-PSNR, higher resolution videos BQTerrace and Traffic has lower degradation in quality. For this video sequences, when ranging from 1 to 10 discarded coefficients BD-PSNR results range from slightly above zero to -0.01 dB. One can realize that some configurations present better BD-PSNR results than the reference precise SATD 4x4. Our hypothesis is that some pruned HT (*e.g.* the 2 discarded coefficient HT) could present slightly better properties than the complete HT in terms of energy concentration. For lower resolution videos the BD-PSNR ranges from slightly above zero to -0.05 dB when considered the 1 discarded coefficient HT to the SAD 4x4.

The BD-BR results are just a complementary evaluation in relation to the BD-PSNR. We show that for higher resolution videos (*i.e.* a) and d)) and considering the range from 1 discarded coefficient up to the SAD 4x4, the BD-BR configuration increases up to 0.4% and 0.5%, respectively. This is an interesting result, since those videos has huge amount of data and do not suffer a large bit-rate overhead when using approximation for the SATD 4x4. For lower resolution videos in b) and c), the increase in BD-BR are not higher than 1%.

C. Energy Efficiency Results

According to the findings in subsection IV-B, we decided to evaluate the energy efficiency for the following approximate architectures configurations: 2, 4, 6, 8, 10 discarded coefficients HT, the SAD 4x4 (*i.e.* 16 discarded coefficients), and the precise SATD 4x4. These pruning choices were evaluated since higher number of discarded coefficients showed worse results in relation to the totally pruned HT configuration of SAD 4x4.

The maximum frequency achieved for each design was 911.3 MHz, 757.2MHz, 741.1 MHz, 675.1MHz, 676.6 MHz, 658.4 MHZ, and 650 MHZ, respectively. The SAD 4x4 and the 10 discarded coefficients SATD 4x4 architectures present reductions in the critical path timing of 28.7% and 14.2% when compared to the SATD 4x4, respectively. In other words, this represents approximate circuits 1.4X and 1.2X faster than the precise one. For the iso-performance analysis, in which all versions clock at the same frequency, we chose the maximum frequency achieved by the precise SATD 4x4, since all the approximate designs can reach this frequency of operation.

Table III shows the iso-performance analysis at the frequency of 650 MHz. Area is presented as the number of NAND2 X1 equivalent gate count (EG). When using approximate designs, the area reduction ranges from 75.9% down to 6.6%. For the 10 discarded coefficients approximate architecture this area reduction reaches 52.75%. The area overhead of this configuration when compared to the SAD 4x4 is of 1.9X, while comparing to the precise SATD 4x4 the area reduction is 2.1X. In terms of energy efficiency, the mean energy per SATD operation (MEOp) shows reductions of 89.5 % for the SAD 4x4 when compared to the precise SATD 4x4 architecture. The approximate architecture with 10 least significant coefficients being discarded presents a significant reduction of 70.7% in energy per operation. When considering the less pruned architectures with 6 and 2 discarded coefficients, the energy per operation has a reduction of 38.5 % and 12.9%.

TABLE III. AREA AND ENERGY EFFICIENCY RESULTS @ 650 MHZ

# of discarded coefficients	Area (EG)	# cells	Leakage Power (µW)	Dynamic Power (µW)	MEOp (pJ)
SAD	3847	2171	74.9	1556.9	2.5
10	7553	3305	146.5	4389.1	7
8	9426	4299	187.2	6214.4	9.9
6	11863	5632	249.8	9305.4	14.7
4	12813	6177	272.5	10216.2	16.2
2	14930	7339	333.2	13160.7	20.8
Precise SATD	15986	8082	354.3	15170.7	23.9

Based on that, one can conclude that our proposed pruning-based algorithm brings substantial energy reduction for the case study on SATD 4x4 and video sequences under evaluation in Table II. Furthermore, for those video sequences we show that good enough video quality results are achieved since the higher BD-PSNR degradation is just -0.05 dB without affecting the compression (i.e. maximum BD-BR of 1%).

V. CONCLUSION

This work proposed a novel pruned-based algorithm for SATD architectures regarding HEVC coding in order to obtain energy-efficient 45nm CMOS accelerator circuits. Our algorithm performs coefficient pruning according to the number of least significant HT coefficients to be discarded. We also show that the SAD architecture can be defined as the case where the entire HT is discarded from the SATD. For the case study on SATD 4x4, results showed good enough video quality with the highest BD-PSNR degradation of just -0.05 dB regarding the lower-bound SAD 4x4 approximate solution. In terms of compression, the increase in BD-BR is minimal, of up to 1%. Area reduction is provided by our resulting approximate architectures with a maximum of 75.9% for SAD 4x4 and 52.75% for the 10 discarded coefficients SATD 4x4 approach. Energy efficiency is high for the latter, with an average of 7 pJ per SATD 4x4 operation while the precise one consumes 23.9 pJ - an energy per operation reduction of 70.7%.

REFERENCES

[1] H. Esmaeilzadeh et al. Dark silicon and the end of multicore scaling. In Computer Architecture (ISCA), 2011 38th Annual International Symposium on, pp 365 –376.

[2] D. Markovic, V. Stojanovic, B. Nikolic, M. A. Horowitz, and R. W. Brodersen, “Methods for true energy-performance optimization,” *IEEE Journal of Solid-State Circuits*, vol. 39, no. 8, pp. 1282–1293, Aug. 2004.

[3] J. Han and M. Orshansky, “Approximate computing: An emerging paradigm for energy-efficient design,” in Test Symposium (ETS), 2013 18th IEEE European, Avignon, 2013, pp. 1–6.

[4] N. Zhu, W. L. Goh, W. Zhang, K. S. Yeo, and K. S. Kong, “Design of Low-Power High-Speed Truncation-Error-Tolerant Adder and Its Application in Digital Signal Processing,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 8, pp. 1225–1229, Aug. 2010.

[5] ITU-T and ISO/IEC, High Efficiency Video Coding, ITU-T Recommendation H.265 and ISO/IEC 23008-2, 2013.

[6] ITU-T and ISO/IEC JTC 1, Advanced video coding, ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), 2011.

[7] J. Vanne et. al. “Comparative Rate-Distortion-Complexity Analysis of HEVC and AVC Video Codecs”. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 22, n. 12, Dec 2012

[8] HEVC Test Model (HM) version 16.0. http://hevc.hhi.fraunhofer.de/

[9] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, “Low-Power Digital Signal Processing Using Approximate Adders,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 124–137, Jan. 2013.

[10] F. Bossen, “Common test conditions and software configurations”, JCT-VC-L1100, JCT-VC, Geneva, Jan. 2013

[11] NanGate 45 nm Open Cell Library, www.nangate.com/?page_id=22.