# Evaluating the Impact of Carry-Ripple and Carry-Lookahead Adders in Pel Decimation VLSI Implementations

**Ismael Seidel, Bruno George de Moraes, José Luís A. Güntzel**
{ismaelseidel, brunogm, guntzel}@inf.ufsc.br

**Embedded Computing Lab. (ECL) – Department of Informatics and Statistics**
**Federal University of Santa Catarina – Florianópolis, Brazil**

## Abstract

*This paper presents an overview concerning the Pel Decimation Algorithm applied to VLSI implementations, briefly describing some novel pel decimation patterns tested in intra prediction. In order to use results of previous adder's logical synthesis, two equations are derivate to estimate the area, critical delay and power consumption costs for complete Sum of Absolutes Differences (SAD). Those equations are based in a simple fully parallel SAD architecture. Using these equations, the synthesis costs are estimated for Carry-Ripple Adder (CRA) and Carry-Lookahead Adder (CLA), which are evaluated regarding energy efficiency for three pel decimation ratios. The border pattern achieves better performance than SAD at the same energy efficiency.*

## 1. Introduction

The huge amount of data in raw digital video makes storage and transmission difficult, if not impossible. Video compression techniques are used to overcome such difficulties. To simplify the compression, each frame of a video sequence is divided into macroblocks, which in turn are further divided into smaller blocks. The blocks constitute the basic pixel matrices that are submitted to Inter frame and Intra frame Prediction techniques, which explore temporal and spatial redundancies, respectively. Both predictions rely on coding the residue data (i.e., the differences) between similar blocks. In such coding, a similarity metric must be used to guide the search for the most similar blocks (i.e., block matching). The Sum of Absolutes Differences (SAD) [1] is the most used similarity metric in VLSI implementations. Basically, it accumulates the module of the residue between the original and the candidate block thus requiring only additions and subtractions. The search for similar blocks is the most complex step in video coding.

There are some proposals to decrease such complexity. One way is to reduce the number of candidates to fit. Another way is to reduce the number of input pixels to the similarity metric, which is known as Pel Decimation [2].

The pel decimation algorithm does not define how the pixels should be sampled over the block. Thus, there are many possible patterns and some of them may result in better candidate choices. In [3] some patterns are proposed for H.264/AVC standard [4] and evaluated in its reference implementation.

When applying pel decimation to SAD, the area, delay and power of the VLSI implementation are determined by the adopted adder and thus, the choice of adder architecture is of utmost importance [5].

In this paper we evaluate the impact of the adder architecture in VLSI implemantion of pel decimated SAD in terms of area, delay, power and energy efficiency (power-delay product - PDP). The considered adder architectures are the Carry-Ripple (CRA) and the Carry-Lookahead (CLA). VLSI implementations for three pel decimation ratios are considered.

The rest of this paper is organized as follows. Section 2 presents the pel decimation algorithm along with subsampling patterns and respective ratios. Section 3 shows the chosen main SAD architecture for the estimates and details how the CRA and CLA costs are derived. Section 4 shows the results whereas Section 5 draws conclusions and outlines prospective works.

## 2. Pel Decimation Sampling Patterns

Thanks to its simplicity, the SAD is widely used for block matching in video coding. According to Lee et al [6], the full-search algorithm, which is used for block matching in inter prediction, gives the best results. This method must compute the SAD for all candidates in a search window. Although each SAD computation is simple and fast, the enormous number of candidates makes full-search prohibitively complex. Such complexity can be alleviated by reducing the number of candidates. This is the reasoning behind the several existing search algorithms (e.g. diamond search).

Another way to reduce complexity is the pel decimation algorithm, which is widely used, like in [2], [6]-[8]. According to Kuhn [2], in pel decimation, some pixels are taken regularly over the search area, resulting in subsampling. This subsampling causes a lower correlation of each pixel in the block. As a result, a decrease

in prediction quality is expected. On the other hand, pel decimation computation exhibits a high regularity, which allied to the achieved complexity reduction makes it very suited for VLSI implementations.

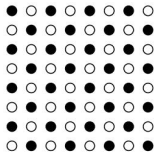

Fig. 1 – Example of 2:1 pel decimation pattern in an 8x8 block. [8]

In pel decimation a regular pattern as the one shown in fig. 1 (in this paper, referred to as "checker") is widely used in the works found in literature. However, the algorithm does not define any standard pattern. In [3] six pel decimation sampling patterns were assessed. Such patterns, showed in fig. 2, use the pel ratios defined in the literature (2:1 and 4:1), except “border” pattern, which sample in 4:3 ratio. This latter pattern includes more block information than usual ratios and still saves about ¼ of the operations compared to full SAD. The results presented in [3] show that for 4x4 intra prediction, “border” (as expected because of its larger ratio) and “cross” patterns have better block match (with less residue) than both “checker” patterns, resulting in better PSNR and SSIM.

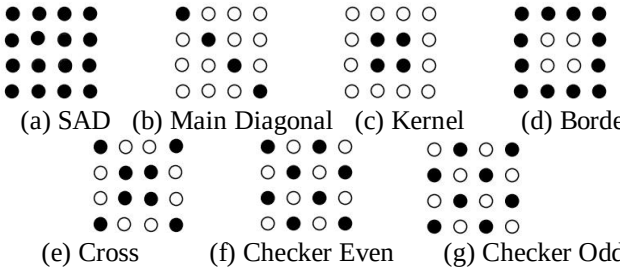


Fig. 2 – SAD (a) and proposed patterns (b)-(g) for 4x4 blocks. The chosen pixels are marked in black. [3]

# 3. Proposed Hardware Architecture for Estimation

Monteiro [9] evaluates some adder architectures, such as the Carry-Ripple (CRA), the Carry-Lookahead (CLA), the Carry-Select (CSA), the Add-One Carry-Select (A1CSA) and the Hierarchical Add-One Carry-Select (A1CSAH). Those adders were synthesized for 45nm TSMC standard-cell library with Synopsys Design Compiler in Topographical mode [10]. The reported results include critical delay, power and area.

Considering 4x4 blocks (as assumed in this work), the maximum bit-width for a SAD is 12 and hence, the CRA and CLA show the best trade-off between performance and power among all evaluated adders in the 8-12 bit range. Considering only these two adders, the CLA presents the shortest critical delay whereas the CRA consumes less power. In this work, only CRA and CLA are assumed. As Monteiro’s synthesis results were reported for adders with bit-widths between 2 and 8, and as long as estimates for 9, 10 and 11 bits were needed, those were obtained by using a non-linear regression. This regression used LABFit [11] software, which searches over 280 functions for the minimum error. Thus, for each adder and each set of area, critical delay and power data, the best adjust function was fit. From these functions it was derived further values, seeking to minimize the estimation errors.

Chen et al [12] present a 720p H.264/AVC coder architecture where the SAD calculation responds for 33% of the total gate count. Liu et al [13], by they turn, present a coder VLSI architecture where 79% of the gates are used to SAD computation. While battery powered portable devices are responding for an ever increasing portion of the electronics market, energy efficiency becomes of prime importance. Therefore, the design of hardware blocks for computing similarity metrics should be optimized for power.

Taking advantage on existing synthesis data for adders (from [9]), this work describes some theoretical equations to estimate the SAD and its pel decimation costs, in terms of delay, power and area. As basis for these equations, we assumed a fully parallel SAD datapath. The basic block, called EP, performs a sum of two absolute differences by parallel processing two pixels of each block (original and reference). This is depicted in fig. 3. The remaining of the datapath is composed by a tree of adders to accumulate the EP outputs for each pel pair. Such approach was shown to be high energy efficiency by Walter, Diniz and Bampi [14]. In that work the most efficient architecture used 16 pixel input and only one pipeline stage, with each pixel being considered with only 8 bits.

In the current work, the datapath is composed by eight EP blocks plus seven adders resulting in a total of 16 subtractors, 16 blocks for extracting the absolute value (ABS) and 15 adders. The critical path can be approximated by a subtractor, an ABS and four adders. The equations 1 and 2 are derived from previous

datapath and critical path components which describes the power and critical delay, in which “n” is the number of pixels in a block.

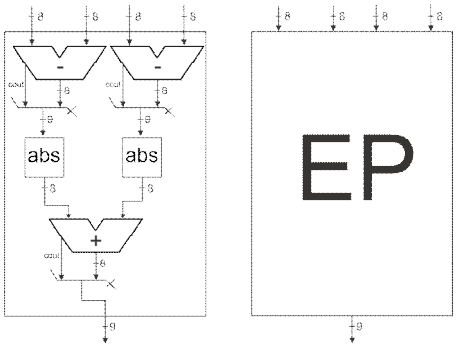


Fig 3 – The “EP” datapath.

$$power_{SAD}(n) = n * power_{subtractor} + n * power_{abs} + (n-1) * power_{adder} \quad (1)$$

$$delay_{SAD}(n) = delay_{subtractor} + delay_{abs} + \log_2 n * delay_{adder} \quad (2)$$

Taking into account that for each level of the tree adders there is a growth of one bit for each adder input of the next level. From this, equation 1 derives equation 3. Further, as the logical synthesis results in [9] are for adders the energy consumption of ABS and subtractors are assumed to be the same as an adder with the same bit-width. Obviously, this kind of simplification reduces the accuracy of the estimates. This way, the inverters at the inputs of a subtractor and the multiplexers at the outputs of the ABS are disregarded, consisting in the main source of error. After such simplification, the power of the datapath can be estimated by Equation 4.

$$power_{SAD}(n) = n * power_{subtractor} + n * power_{abs} + \sum_{i=0}^{\log_2 n-1} \left(2^i power_{adder(7+\log_2 n-i)}\right) \quad (3)$$

$$power_{SAD}(n) = \frac{5n}{2} * power_{adder(8)} + \sum_{i=0}^{\log_2 n-2} \left(2^i power_{adder(7+\log_2 n-i)}\right) + error \quad (4)$$

Similarly the critical delay should take into account the bit growth in the adders tree, and the substitution of subtractors and absolute extractors by adders. Thus, equation 5 can be derived from equation 2.

$$delay_{SAD}(n) = 3 * delay_{adder(8)} + \sum_{i=0}^{\log_2 n-2} delay_{adder(9+i)} + error \quad (5)$$

## 4. Delay, power and area estimates

Tab.1 shows critical delay, power and area for (full) SAD and each considered pel decimation ratio, assuming both CRA and CLA. The area was derived by using equation 4, replacing power by area. Also, the Power-Delay Product (PDP) was included to estimate the energy efficiency. Thereby, the total power used is in accordance with experimentations made by Monteiro [9]. Fig. 3 shows the PDP and Area values of tab. 1 in graphic format.

Tab.1 – Estimates of studied sampling patterns in [3] using CLA and CRA results by [9].

| **Metric** | **Critical Delay (ps)** | | **Power (µW)** | | **Area (µm²)** | | **PDP (pJ)** | |
|---|---|---|---|---|---|---|---|---|
| | CLA | CRA | CLA | CRA | CLA | CRA | CLA | CRA |
| SAD | 2045,37 | 2513,45 | 1775,5 | 1129,99 | 4074,40 | 1706,67 | 3,59 | 2,84 |
| Border | 2045,37 | 2513,45 | 1306,55 | 840,95 | 3031,41 | 1270,08 | 2,67 | 2,11 |
| 2:1 | 1654,49 | 2087,02 | 845,9 | 549,14 | 1970,84 | 829,08 | 1,4 | 1,15 |
| 4:1 | 1288,90 | 1650,22 | 396,95 | 260,1 | 927,86 | 392,49 | 0,51 | 0,43 |

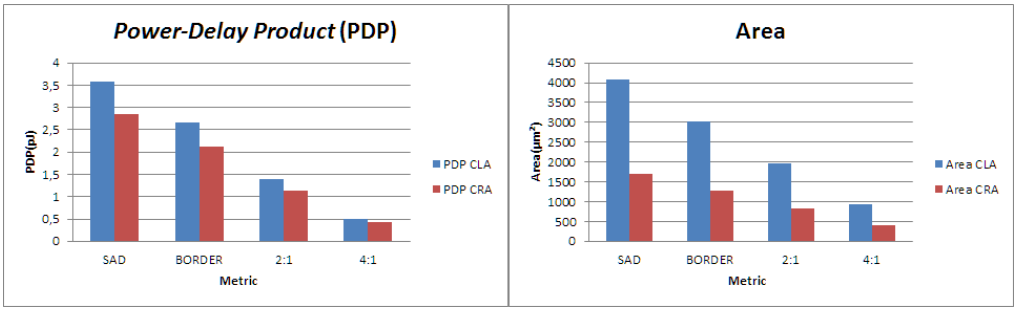


Fig. 3 – PDP (pJ) and Area (μm²) CLA and CRA results for each subsampling pattern.

## 5. Conclusions and Future Works

This paper presented six pel decimation patterns (including four novel patterns proposed in [3]), as well derivate equations to estimate the costs of a hardware implementation of a Sum of Absolutes Differences (SAD) block using previous adders synthesis results.

The Power-Delay Product (PDP) differences between CLA and CRA decreases with the pel sampling ratio. The Carry-Ripple Adder (CRA) implementation is about 1,22x as energy efficient as the Carry-Lookahead Adder (CLA). As expected, the most efficient pel decimation proportion is the 4:1 that uses only 25% of the block. The area estimates between subsampling ratios changes less in CRA than in CLA. The critical delay for the full SAD and "border" pattern are the same, because of the same tree adder height for accumulating the absolutes differences for the same adder are equal. Also, the "border" pattern with CLA has the same PDP than SAD with CRA which means that "border" will be faster than SAD, while providing the same energy efficiency.

Prospective works include the circuit description in a Hardware Description Language (HDL), and the proper synthesis. Also, new pel decimation patterns are being tested with the x264 [15] encoder, once its speed provides for longer and embracing tests, making possible a statistical analysis of pel decimation pattern impacts on video coding quality.

## 6. References

[1] I. Richardson, H.264 and MPEG-4 video compression – Video Coding for Next-Generation Multimedia. John Wiley&Sons, Chichester, 2003.

[2] P. M. Kuhn, "Fast mpeg-4 motion estimation: Proces-sor based and flexible vlsi implementations," The Journal of VLSI Signal Processing, vol. 23, pp. 67–92, 1999, 10.1023/A:1008192719838.

[3] I. Seidel. "Investigação de padrões de subamostragem em métrica de similaridade de imagem". Final Undergraduate Project, UFSC, Florianópolis, 2011.

[4] ITU-T, "H.264 corrigendum 1," jan 2009.

[5] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, Digital Integrated Circuits, 2nd ed. Prentice Hall, 2003.

[6] Kun-Bin Lee, Hao-Yun Chin, Hui-Cheng Hsu, and Chein-Wei Jen, "QME: an efficient subsampling-based block matching algorithm for motion estimation", in Proceedings of the 2004 International Symposium on Circuits and Systems, 2004. ISCAS '04, 2004, vol. 2, p. II– 305–8 Vol.2.

[7] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors", IEEE Transactions on Circuits and Systems for Video Technology, vol. 3, nº. 2, p. 148–157, abr. 1993.

[8] M. S. Porto, "Investigação de algoritmos e desenvolvimento arquitetural para estimação de movimento em compressão de vídeo digital", Final Undergraduate Project, UFPel, 2006.

[9] J. L. Monteiro. "Hierarchical Add-One Carry-Select Adder: Um Somador Select-Adder com Cadeia de Carry Logarítmica". Final Undergraduate Project, UFSC, Florianópolis, 2011.

[10] Synopsys. Synopsys's Design Compiler User Guide, Version C-2009.06. 2011.

[11] LABFit. LAB Fit Curve Fitting Software. 2011.

[12] T.-C. Chen, S.-Y. Chien, Yu-Wen Huang, Chen-Han Tsai, Ching-Yeh Chen, To-Wei Chen, and Liang-Gee Chen, "Analysis and architecture design of an HDTV720p 30 frames/s H.264/AVC encoder", IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, nº. 6, p. 673–688, jun. 2006.

[13] Z. Liu, Y. Song, M. Shao, S. Li, L. Li, S. Goto, and T. Ikenaga, "32-Parallel SAD Tree Hardwired Engine for Variable Block Size Motion Estimation in HDTV1080P Real-Time Encoding Application", in 2007 IEEE Workshop on Signal Processing Systems, 2007, p. 675–680.

[14] F. L. Walter, C. M. Diniz, and S. Bampi, "Synthesis and comparison of low-power high-throughput architectures for SAD calculation", in 2011 IEEE Second Latin American Symposium on Circuits and Systems (LASCAS), 2011, p. 1–4.

[15] L. Merritt, J. Garrett-Glaser, S. Walters, A. Mitrofanov, H. Gramner, and D. Kang, “x264 project,” 2004.