A VLSI IMPLEMENTATION FOR FAST ALL-BOOLEAN MOTION ESTIMATION BASED ON PRE-CODED IMAGE PLANES MATCHING

Anıl Çelebi Sarp Ertürk Ali Tangel
e-mail: anilcelebi@kou.edu.tr e-mail: sertur@kou.edu.tr e-mail: atangel@kou.edu.tr
Kocaeli University, Faculty of Engineering, Department of Electronics & Telecommunications Engineering, 41040, İzmit, Kocaeli, Turkey

Key words: Motion Estimation, High-Speed VLSI

ABSTRACT

This paper presents a VLSI implementation of a new approach for block based motion estimation based on Boolean matching in pre-coded image planes. The implemented system adds redundancy to the pixel representation for the purpose of motion estimation computation reduction. Images are pre-coded to enable a Boolean-only matching criterion that approximates the mean absolute difference. Fast matching criteria evaluation with a simple scheme suitable for hardware implementation is facilitated. The implemented VSLI system is optimized for high speed.

I. INTRODUCTION

Motion estimation (ME) and compensation play key roles in video coding systems due to the ability of realizing high compression rates achieved by reducing temporal redundancies between successive image frames. Motion estimation is the most computationally intensive part of the video coding system, performing up to 50 % of the computations executed by the entire coding system [1]. The most popular technique for motion estimation is block matching using the full search (FS) algorithm with mean absolute difference (MAD) or mean square error (MSE) matching criteria. Various methods have been proposed to reduce the computational load of the full-search block matching algorithm. Proposed approaches can be divided into three categories [2]: fast search techniques that select a subset of the possible candidate locations, techniques based on various forms of pixel pattern decimation that employ a certain sub-sampling pixel pattern, and techniques that exploit different matching criteria instead of the classical MAD.

Techniques that exploit different matching criteria have been proposed to achieve reduction in computational complexity by simpler evaluation criteria instead of the MAD. Motion estimation by matching of image bit-planes [3], hierarchical feature matching-motion estimation (HFM-ME) which employs sign truncated feature (STF) matching [4], and motion estimation using the one-bit transform (1BT) or the modified one-bit transform [5] are several examples of techniques proposed for a simpler matching criteria.

In this paper, a novel approach for motion estimation by Boolean block-matching using pre-coding of image frames is proposed. By adding redundancy to the pixel representation, a reduction in the computational load of matching criteria evaluation (compared to MAD) for motion estimation is achieved. Images are pre-coded to achieve improved motion estimation performance by Boolean only block matching, the coding enabling a close approximation to the MAD criterion. Coding of image frames is performed only once for each frame while computations for the best match are executed exhaustively, thus the reduced computational load of the matching criterion provides a gain in computation time considering the entire motion estimation process.

II. BLOCK MOTION ESTIMATION BY PRE-CODED IMAGE-PLANE MATCHING

The grey level of a pixel at location *(x, y)* in the *t*-th image frame, in a sequence with image bit-depth K, can be represented as

$$f^t(x,y) = a_{K-1}2^{K-1} + a_{K-2}2^{K-2} + \cdots + a_1 2^1 + a_0 2^0 \quad (1)$$

where a_k, $0 \le k \le K$ is either 0 or 1. Block motion estimation based on bit-plane matching (BPM) evaluates the matching performance according to bit-plane correlation evaluated between the bit-planes of the current frame and the previous frame for each possible block displacement, using the Boolean exclusive-OR operation. The total difference measure for each motion vector candidate is obtained by weighting bit-plane correlations according to their level of significance. The reason for a degraded matching accuracy of bit-plane matching compared to the MAD criterion, is that close pixel values can result in fairly different bit-planes so that a larger

difference measure can be introduced in BPM than it is actually the case.

Gray-coding can be used to reduce small grey-level variations in bit-planes: pixel values are represented by their corresponding Gray code where successive code-words differ in only one bit position. The *K*-bit Gray code $g_{K-1}\cdots g_2 g_1 g_0$ can be formulated as

$$\begin{aligned} g_{K-1} &= a_{K-1} \\ g_k &= a_k \oplus a_{k+1} \ \text{ for } \ 0 \le k \le K-2 \end{aligned} \tag{2}$$

The *k*-th order Gray-coded bit-plane $g_k^t(x,y)$ of an image frame is defined to contain all the *k*-th order Gray coded bits g_k of the image. Gray-coded bit plane matching (GC-BPM) has for instance been proposed for global motion estimation in digital image stabilizers [6] to improve on the accuracy of direct bit-plane matching.

Figure 1 shows the assignment of the four most significant bits of the Gray-code according to the pixel range 0-255 of an 8 bit image representation. Gray-coding improves motion estimation performance compared to direct BPM. However, the block motion estimation performance of GC-BPM is still inferior to MAD due to two reasons: firstly pixels near in value can be Gray-coded into separate bit-planes at higher levels (at the more significant bit-planes) introducing a larger difference measure than it should be, and secondly pixels relatively far in value can be Gray-coded into similar bit-planes introducing a smaller difference measure than it should be.

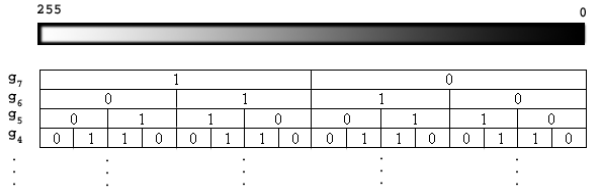


Figure 1. Assignment of the four most significant bits of the Gray-code according to the pixel range 0-255 of an 8 bit image representation.

Block motion estimation by pre-coded image-plane matching is proposed to add redundancy to the pixel representation for the purpose of closely approximating the actual distance between pixel values by the EX-OR distance of coded bit-planes [7]. By increasing the number of bits used to represent pixel values it is possible to avoid close pixel values introducing a difference measure as a result of the EX-OR computation. Figure 2 shows the top levels of an example code carrying the desired features. The code employs three bits for the representation of the top most level. This procedure makes possible that no difference is introduced for pixels falling into the same field as well as pixels falling into adjacent fields (i.e. pixels that differ in none or only in one of the three bits), while pixel values further apart are ensured to differ in two or three bits according to their distance to each other. Increasing the number of bits used at each level, it is facilitated that only two bit differences will introduce a weighted distance so that adjacent fields do not get a distance measure introduced. Additionally, auxiliary levels are utilized to introduce a distance measure for far pixel values (solving the second problem of GC-BPM).

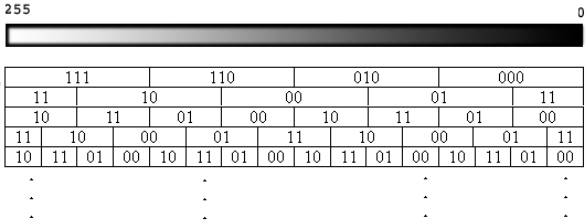


Figure 2. Assignment of the top significant bits of a sample PCIPM code according to the pixel range 0-255 of an 8 bit image representation.

The resultant codebook for an example code that can be used to represent an 8-bit image to satisfy the requirements, consisting of a 27 bit representation is obtained through the following coding,

$$\begin{aligned}
 & & c_{13} &= a_4 \\
c_{26} &= a_7 & c_{12} &= g_3 \\
c_{25} &= a_7 + a_6 & c_{11} &= g_4 \cdot g_2 + g'_3 \cdot g'_2 \\
c_{24} &= a_7 \cdot a_6 & c_{10} &= g'_4 \cdot g_2 + g'_3 \cdot g'_2 \\
c_{23} &= g_7 \cdot g_5 + g'_6 \cdot g'_5 & c_{09} &= a_3 \\
c_{22} &= g'_7 \cdot g_5 + g'_6 \cdot g'_5 & c_{08} &= g_2 \\
c_{21} &= a_6 & c_{07} &= g_3 \cdot g_1 + g'_2 \cdot g'_1 \\
c_{20} &= g_5 & c_{06} &= g'_3 \cdot g_1 + g'_2 \cdot g'_1 \\
c_{19} &= g_6 \cdot g_4 + g'_5 \cdot g'_4 & c_{05} &= a_2 \\
c_{18} &= g'_6 \cdot g_4 + g'_5 \cdot g'_4 & c_{04} &= g_1 \\
c_{17} &= a_5 & c_{03} &= g_2 \cdot g_0 + g'_1 \cdot g'_0 \\
c_{16} &= g_4 & c_{02} &= g'_2 \cdot g_0 + g'_1 \cdot g'_0 \\
c_{15} &= g_5 \cdot g_3 + g'_4 \cdot g'_3 & c_{01} &= a_1 \\
c_{14} &= g'_5 \cdot g_3 + g'_4 \cdot g'_3 & c_{00} &= g_0
\end{aligned}$$

where a_k corresponds to the *k*-th bit of the pixel in the original image and g_k corresponds to the *k*-th bit of the Gray code of the pixel. "·" denotes Boolean AND, "+" denotes Boolean OR operation. Only the auxiliary levels require a slightly more complex Boolean arithmetic. For the matching process, each image frame is pre-coded using the corresponding coding scheme and pre-coded image plane matching (PCIPM) motion estimation is carried out by simply weighting more than one bit differences at each level according to their level of significance.

Although the number of bits required per pixel for the pre-coding of images is considerably increased, this addition of redundancy enables more accurate as well as fast EX-OR matching evaluation. Because each image frame has to be pre-coded only once, while the motion estimation criteria is repeatedly executed for each block candidate position, the simplification into a Boolean only motion estimation system is worth the pre-coding overhead.

III. VLSI IMPLEMENTATION

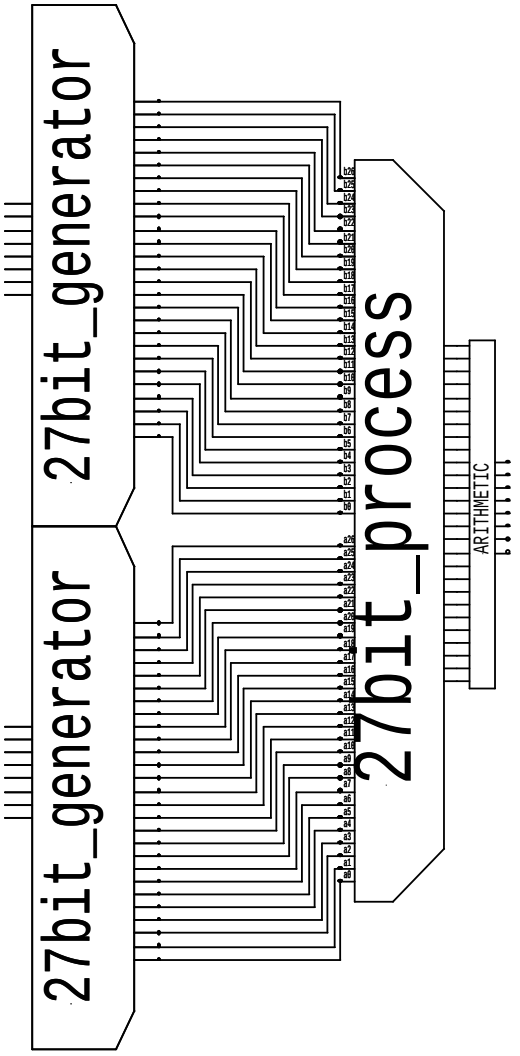


Figure 3. Block diagram of the system to be implemented

In Figure 3, the block labelled “27bit_generator” is the system that implements the encoding logic described in part II of which the interior is shown in Figure 5. In addition, the system needs an 8-bit adder circuitry in the arithmetic block. The arithmetic block and the utilized adder circuitry are shown in Figure 6 including the sub block schematics. According to the simulation results based on 0.5micron level 3 CMOS process parameters, the total delay time for the whole system is only 6 *ns* so a clock cycle time of 6 ns can be used for this architecture. As a result, complex and consecutive logic operations become possible to implement, leading to real time applications. A larger layout area is needed to meet high gain transistor requirements. The entire system uses a total of 1558 transistors, with 17 different geometries.

The proposed system is the most important unit of the motion estimation algorithm because these blocks represent the new approach to the motion estimation task. The 27-bit generator block encodes the 8-bit pixel value to a 27-bit number, the 27-bit process block executes the respective ex-or process, and finally the arithmetic unit, rebuilds the 8-bit difference data from the output of the 27-bit process block. The adder used in the arithmetic block increases the total delay of the system. A typical response of this structure to an example input data is shown in Figure 4. One pixel value is set to “00000000” for reference and the outputs for test pixel values of “11111111”, “00000000”, and “11001100” are evaluated. The period of each bit is 20ns.

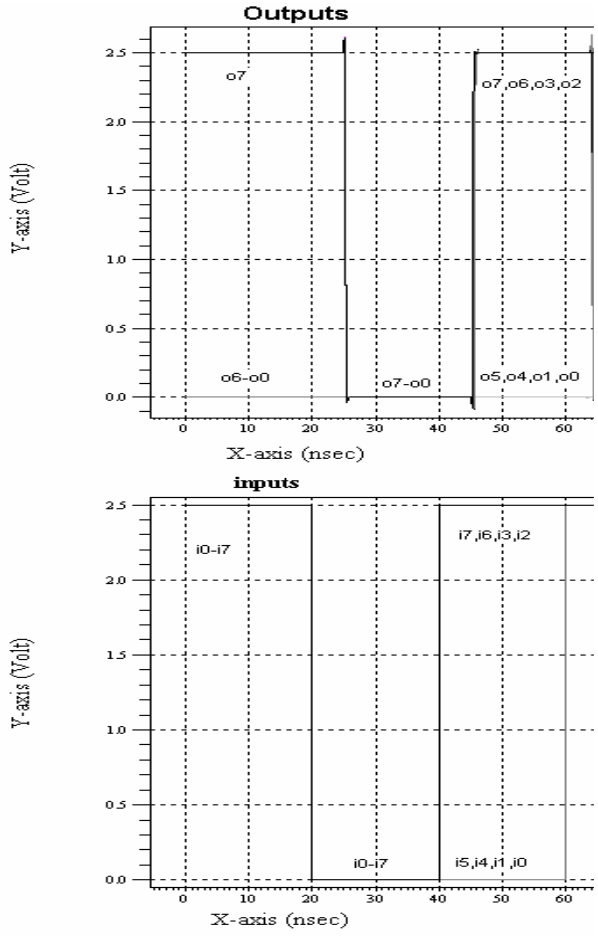


Figure 4. The simulation results for a sample input pair.

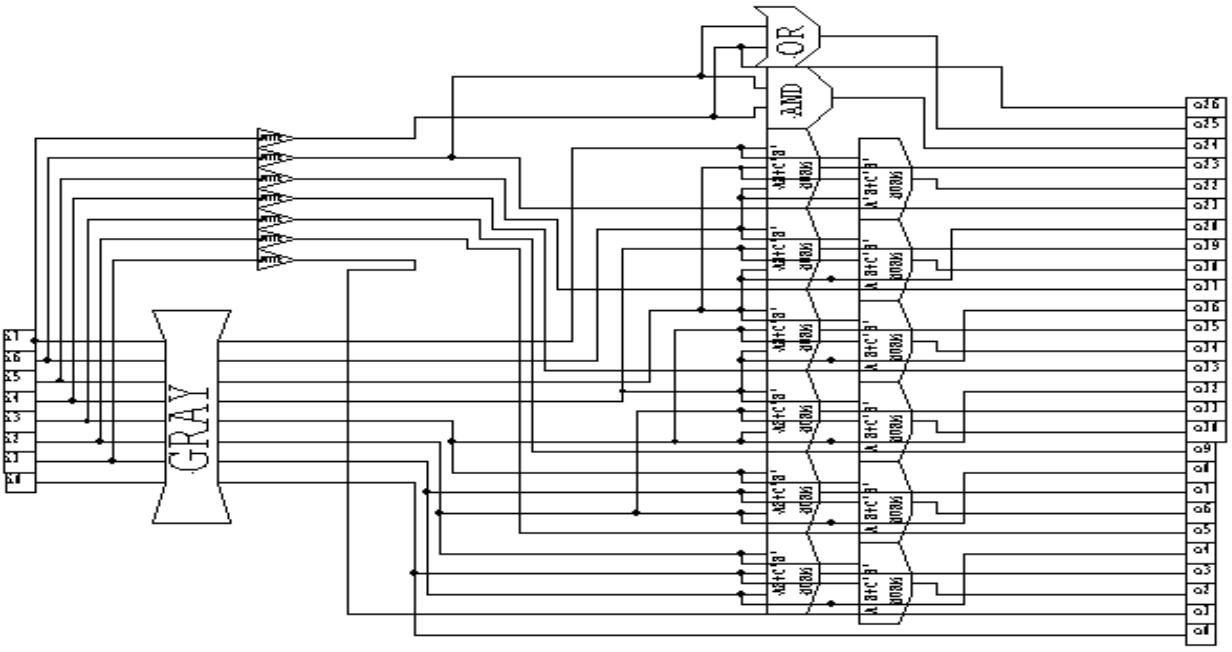
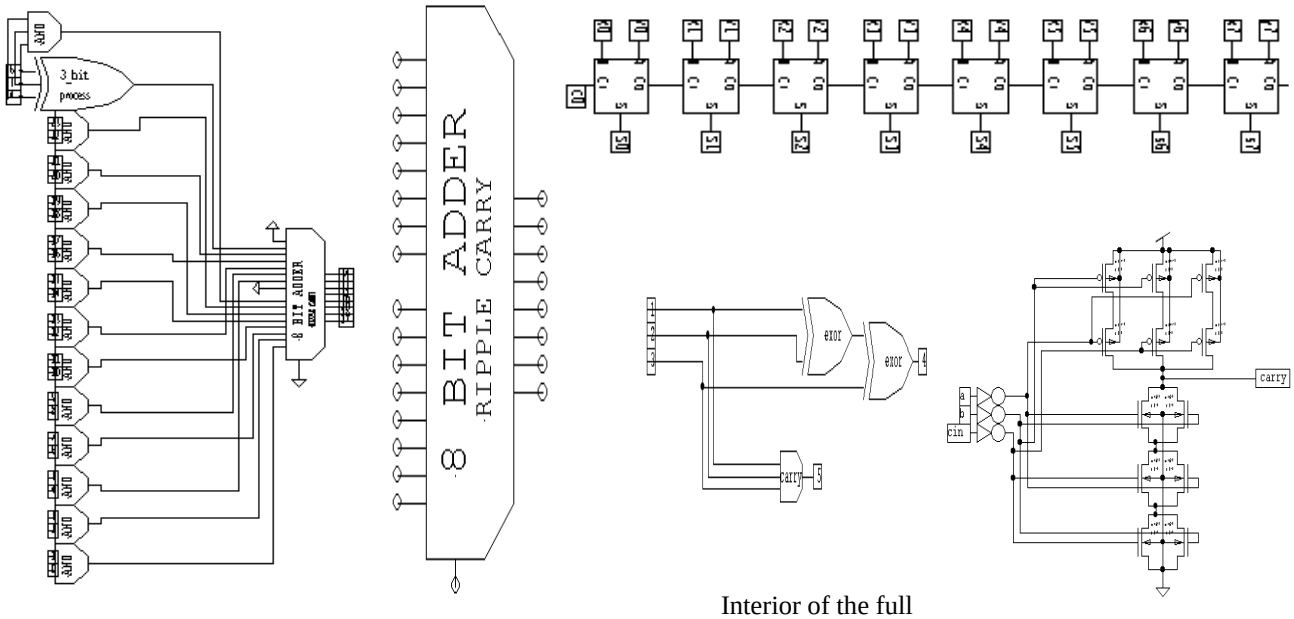


Figure 5. Interior of the block diagram of the 27bit_generator



Interior of the full adder block

Interior of the carry block

Figure 6. The “arithmetic” block and “8-bit ripple carry adder” block diagram.

V. CONCLUSION

This paper presents the VLSI architecture for a novel all Boolean based motion estimation unit designed to estimate interframe image motions. To enable Boolean only motion estimation, the original 8-bit pixel representation is initially encoded into 27 bits. A simple ex-or evaluation between the 27 bits of two pixels that are checked for correspondence is carried out for the matching process. The 27 bit ex-or result is then mapped into two 8 bit difference results approximating the absolute difference between pixel values.

REFERENCES

1. Z.-L. He, C.-Y. Tsui, K.-K. Chan, and M.L.Liou, “Low-power VLSI design for motion estimation using adaptive pixel truncation,” IEEE Trans. Circuits Sys. Video Tech., vol.10, pp. 669-678, Aug. 2000.
2. M. Mattavelli, and G. Zoia: “Vector-tracing algorithm for motion estimation in large search windows,” IEEE Trans. Circuits Sys. Video Tech., vol.10, pp. 1426-1437, Dec. 2000.
3. J. Feng, K.-T. Lo, H. Mehrpour, A. E. Karbowiak, “Adaptive block matching motion estimation algorithm using bit-plane matching,” in Proc. ICIP’95, pp. 496 499, 1995.
4. X. Lee and Y. Zhang, "A Fast Hierarchical Motion-Compensation Scheme for Video Coding Using Block Feature Matching," IEEE Trans. Circuit Sys. Video Tech., vol. 6, pp. 627 - 635, Dec. 1996.
5. P. H. W. Wong and O. C. Au, "Modified One-Bit Transform for Motion Estimation," IEEE Trans. Circuit Sys. Video Tech., vol. 9, pp. 1020 - 1024, Oct. 1999.
6. S.-J. Ko, S.-H. Lee, S.-W. Jeon, and E.-S. Kong, “Fast digital image stabilizer based on Gray-coded bit-plane matching“, IEEE Trans. Consumer Electronics., vol. 45, pp. 598-603, Aug. 1999.
7. S. Ertürk: “Motion estimation by pre-coded image planes matching”, Proc. IEEE International Conference on Image Processing, DOI 0-7803-7750-8/03/, 2003.