A High Performance Hardware Architecture for One Bit Transform Based Motion Estimation

Abdulkadir Akin, Yigit Dogan, and Ilker Hamzaoglu

Faculty of Engineering and Natural Sciences, Sabanci University
34956, Tuzla, Istanbul, Turkey
abdulkadir@su.sabanciuniv.edu, yigitdogan@su.sabanciuniv.edu, hamzaoglu@sabanciuniv.edu

Abstract

Motion Estimation (ME) is the most computationally intensive part of video compression and video enhancement systems. One bit transform (1BT) based ME algorithms have low computational complexity. Therefore, in this paper, we propose a high performance systolic hardware architecture for 1BT based ME. The proposed hardware performs full search ME for 4 Macroblocks in parallel and it is the fastest 1BT based ME hardware reported in the literature. In addition, it uses less on-chip memory than the previous 1BT based ME hardware by using a novel data reuse scheme and memory organization. The proposed hardware is implemented in Verilog HDL. It consumes %34 of the slices in a Xilinx XC2VP30-7 FPGA. It works at 115 MHz in the same FPGA and is capable of processing 50 1920x1080 full High Definition frames per second. Therefore, it can be used in consumer electronics products that require real-time video processing or compression.

Keywords

Motion Estimation, One-Bit Transform, Hardware Implementation, FPGA

1. INTRODUCTION

Motion Estimation (ME) is the most computationally intensive part of video compression and video enhancement systems. ME is used to reduce the bit-rate in video compression systems by exploiting the temporal redundancy between successive frames, and it is used to enhance the quality of displayed images in video enhancement systems by extracting the true motion information. ME is used in video compression standards such as MPEG4 and H.264 [1], and in video enhancement algorithms such as frame rate conversion [2-3].

Block Matching (BM) is the most preferred method for ME. BM partitions current frame into non-overlapping NxN rectangular blocks and tries to find a block from a reference frame in a given search range that best matches the current block. Sum of Absolute Differences (SAD) is the most preferred block matching criterion. The SAD value of a search location defined by the motion vector $d(d_x,d_y)$ is calculated as in (1), where $c(x,y)$ and $r(x,y)$ represent current and reference frames, respectively. The coordinates (i,j) denote the offset locations of current and reference blocks of size NxN.

$$SAD(\vec{d}) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\left|c(x+i,y+j) - r(x+i+d_x,y+j+d_y)\right| \quad (1)$$

Among the BM algorithms, Full Search (FS) algorithm achieves the best performance since it searches all search locations in a given search range. However, the computational complexity of FS algorithm is very high.

Several fast search ME algorithms, such as New Three Step Search (NTSS) [4], Diamond Search (DS) [5], Hexagon-Based Search (HEXBS) [6], and Adaptive Dual Cross Search (ADCS) [7], are proposed to reduce the computational complexity of FS algorithm. These algorithms try to approach the PSNR of FS algorithm by computing the SAD values for fewer search locations in a given search range. Several hardware architectures for fast search ME algorithms are proposed in the literature [8, 9].

Another preferred method for reducing the computational complexity of FS algorithm is reducing pixel resolution from 8 bits to fewer bits. In [10], the one-bit transform (1BT) technique is proposed to reduce the computational complexity of the matching process in ME by transforming video frames into 1 bit/pixel representations and performing ME using these binary representations. Although an 8-bit SAD calculation requires a subtraction and absolute value operation, 1-bit matching only requires an exclusive-or (XOR) operation and is very suitable for hardware implementation.

In [10], video frames are filtered using a multi-bandpass filter and the filtered results are used as pixel-wise thresholds to construct the binary representations used for ME. In [11], a new multi-bandpass filter kernel is proposed for 1BT to facilitate a multiplication free transform for reduced transform complexity. An early termination scheme for binary ME is presented in [12]. In [13], two bit transform (2BT) is proposed to improve ME accuracy compared to 1BT by constructing two bit-planes for each video frame and performing ME using 2BT representations. In [14], constraint one-bit transform (C-1BT) is proposed and it is shown that C-1BT provides increased ME accuracy compared to 2BT at much lower complexity.

The first 1BT based ME hardware implementation in the literature is presented in [10]. In [10], a motion vector (MV) based linear arrays hardware architecture is used for implementing 1BT based ME. In [15], a source pixel based linear arrays hardware architecture is used for implementing low bit depth ME algorithms proposed in [11] and [14]. In [16], a new sub-pixel accurate low bit depth ME algorithm and its hardware is presented.

Data reuse method is used for reducing the off-chip and on-chip memory bandwidth required by ME hardware [17, 18]. In [17], a redundancy access factor and four levels of data reuse are provided to measure the degree of redundancy and a FS hardware architecture with a low memory bandwidth is proposed. In [18], search window (SW) data flows both horizontally and vertically while the current block is stationary and processing element design ensures highly regular dataflow through processing elements avoiding long interconnect delays.

In this paper, we propose a high performance systolic hardware architecture for 1BT based ME. The proposed 1BT ME hardware architecture is based on the 8 bits/pixel ME hardware architectures proposed in [8, 9]. The major differences between them are the proposed ME hardware calculates MVs of 4 Macroblocks (MBs) in parallel, uses a novel data reuse scheme, and uses less memory, processing element array and adder tree area because of 1BT.

The proposed hardware is the fastest 1BT based ME hardware reported in the literature and it is capable of processing 1920x1080 full High Definition (HD) videos in real-time. The 1BT based ME hardware proposed in [10] and [15] cannot process 1920x1080 full HD videos in real-time. The Non-Match Counter architecture used in the proposed ME hardware is faster and has smaller area than the Non-Match Counter architecture used in the ME hardware proposed in [10] and [15]. Although the proposed ME hardware stores search windows of 4 MBs in on-chip memory, it uses less on-chip memory and it loads the on-chip memory from off-chip memory in less number of clock cycles than the ME hardware proposed in [10] and [15].

The proposed hardware is implemented in Verilog HDL. The Verilog RTL code is verified by simulation using Mentor Graphics Modelsim and is mapped to Xilinx *XC2VP30-7* FPGA using Xilinx ISE. It consumes 4758 slices (34% of all the slices) and 8 BlockRAMs (BRAMs). It works at 115 MHz and is capable of processing 50 1920x1080 full HD frames per second. Therefore, the proposed 1BT ME hardware can be used in consumer electronics products that require real-time video processing or compression.

The rest of the paper is organized as follows. Section 2 explains 1BT based ME. In Section 3, the systolic hardware architecture for 1BT based ME is described and its implementation results are given. Section 4 concludes the paper.

2. 1BT BASED MOTION ESTIMATION

In [10], a multi-band pass filter that has 25 non-zero elements is used to obtain filtered images. The filtered images are compared to the original images to create the one-bit images. In this case, non-integer operations are required for the normalization stage of the filtering which has comparatively higher computational complexity. In [11], a novel diamond shape kernel (2) is proposed to decrease the computational complexity of the filtering stage of 1BT. This new kernel contains a power of 2 non-zero elements and thus the multiplication operation becomes simple logical shift. For filtering, sixteen additions and one four-bit shift operation is performed.

$$K=\frac{1}{16}\begin{bmatrix}
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0
\end{bmatrix} \quad (2)$$

$$B(i,j)=\begin{cases}1, & if\ I(i,j)\geq I_F(i,j)\\ 0, & otherwise\end{cases} \quad (3)$$

$$NNMP(m,n)=\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}B^{t}(i,j)\oplus B^{t-1}(i+m,j+n) \quad (4)$$

Therefore, this method is called multiplication free one-bit transform (MF1BT). In MF1BT, the standard bit-plane is obtained as in conventional 1BT as shown in (3). The number of non-matching points (NNMP) criterion proposed in [10] is then used to evaluate the match of two blocks as shown in (4). The symbol $\oplus$ denotes XOR operation. The search location which has the smallest *NNMP* value is selected as the MV of the current block.

The high performance systolic ME hardware proposed in this paper implements both the 1BT based ME proposed in [10] and MF1BT based ME proposed in [11].

3. PROPOSED HARDWARE ARCHITECTURE FOR 1BT BASED MOTION ESTIMATION

The block diagram of the proposed 1BT based ME hardware is shown in Figure 1. The hardware has 8 BRAMs, Vertical Rotator, One Bit Selector, 4 Processing Element (PE) arrays, Control Unit, Comparator & MV Generator. The hardware finds MVs of 4 16x16 MBs in parallel using full search ME algorithm based on minimum NNMP criterion in a search range of [-16, 16] pixels. Its latency is 6 clock cycles; one cycle for synchronous read from memory, one cycle for Vertical Rotator and One Bit

Selector, one cycle for Non-Match Counter, two cycles for Adder Tree and one cycle for Comparator & MV Generator. The Control Unit generates the required address and control signals to compute the NNMP values of the search locations in the search windows of the 4 MBs.

Search windows of 4 16x16 MBs (MB0, MB1, MB2 and MB3) and their search locations for [0,0] MV are shown in Figure 2. Total search window (SW_T) size for 4 MBs is 48x96 pixels. There are large intersections between the search windows of these MBs, e.g. 2/3 of the SWs of MB3 and MB2 are the same and 1/3 of the SWs of MB3 and MB1 are the same. Therefore, performing ME for these 4 MBs in parallel allows significant data reuse.

The search locations in a SW_T are searched line by line and the search locations in each line are searched from right to left. MB PE arrays start at the same time by searching their right most search locations in the first line of the SW_T and finish at the same time after searching their left most search locations in the line 32 of the SW_T. The first search location searched by MB3 includes the SW pixels 0 to 15 in the lines 0 to 15. The first search locations searched by the other three MBs include the SW pixels 16 to 31, 32 to 47, 48 to 63 respectively in the lines 0 to 15. After MB PE arrays finish searching their left most search locations in a line, they search the search locations in the next line of the SW_T starting from their right most search locations in that line.

Comparator & MV Generator compares the NNMP values computed by each PE array and determines the minimum NNMP value and the corresponding MV for each MB (MB0, MB1, MB2 and MB3).

A. Systolic PE Array and Data Reuse Scheme

There are 256 PEs in each PE array. The architecture of MB1 PE array is shown in Figure 3. After a PE array computes the NNMP value for a search location in a line, it computes the NNMP value for the search location one pixel left in the same line. The SW pixels needed for computing the NNMP values for first search locations of 4 MBs in a line are loaded from BRAMs into PE arrays. PE arrays, then, reuse SW pixels for computing the NNMP values for the neighboring search locations in a line. The same current MB pixel is used by a PE while computing NNMP values for $(16+16+1)^2 = 1089$ search locations.

Each PE is connected to its neighboring PE in order to shift the SW pixel to right by one. Therefore, each PE array needs 16 new SW pixels for computing the NNMP value for the next search location. The 16 new SW pixels needed by MB3 PE array for computing the NNMP value for the second search location in the first line are pixel 16 in the lines 0 to 15. MB3 PE array gets the 16 new SW pixels it needs from MB2 PE array. Similarly, MB2 PE array gets the 16 new SW pixels it needs from MB1 PE array and MB1 PE array gets the 16 new SW pixels it needs from MB0 PE array. MB0 PE array gets the 16 new SW pixels from One Bit Selector.

The architecture of a PE is shown in Figure 4. Each PE performs an XOR operation between a SW pixel and a current MB pixel. The result of the XOR operation indicates whether these pixels match or not. The results of the XOR operations performed by all 256 PEs in a PE Array for a search location in the SW should be added to compute the NNMP value for that search location. In the proposed architecture, in order to compute the NNMP value for a search location, first NNMP values for each row of the MB are computed by using Non-Match Counters, then the results of these 16 Non-Match Counters are added by an Adder Tree. Therefore, in the proposed hardware, there are 64 Non-Match Counters and 4 Adder Trees.

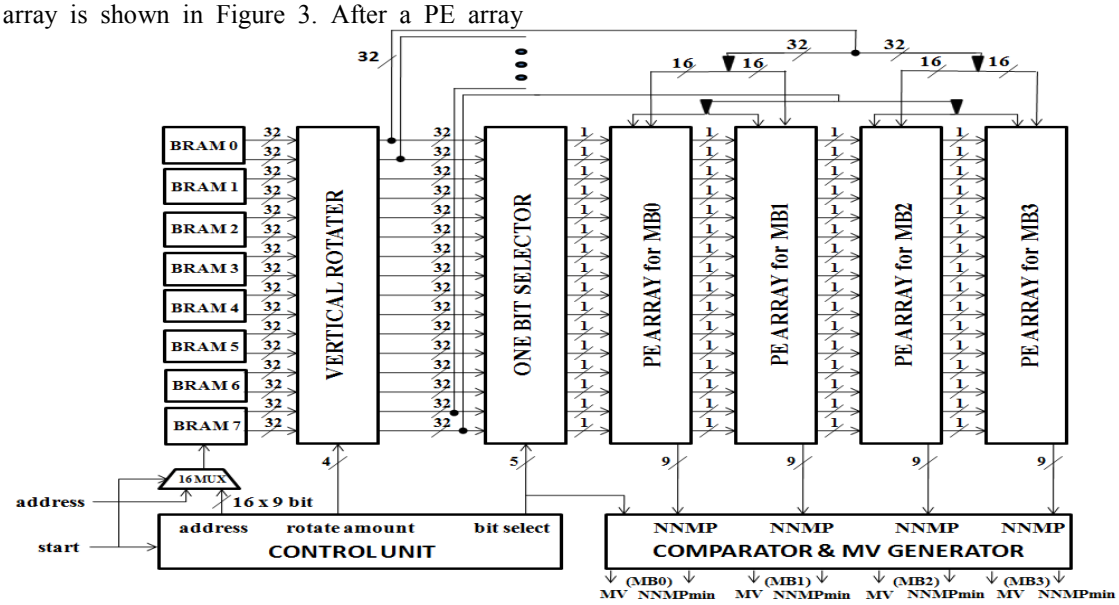


Figure 1. Top-level Block Diagram of Proposed ME Hardware

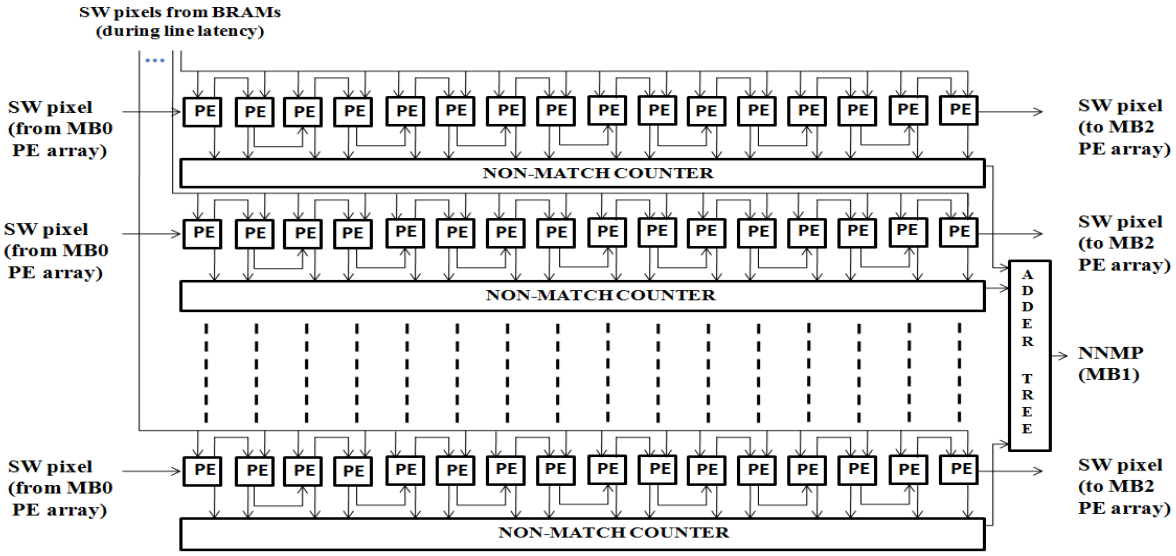


Figure 3. MB1 PE Array

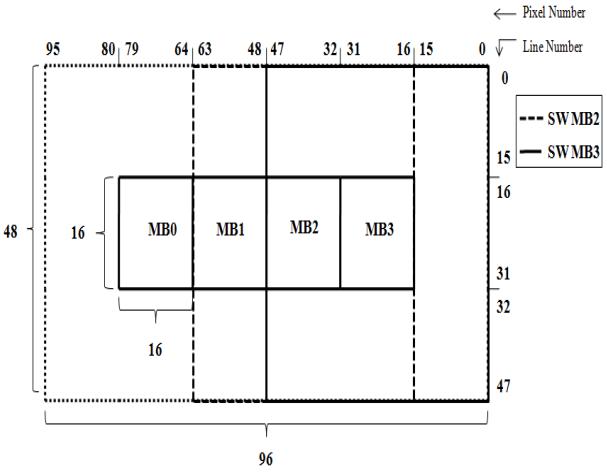


Figure 2. Search Windows of 4 Macroblocks

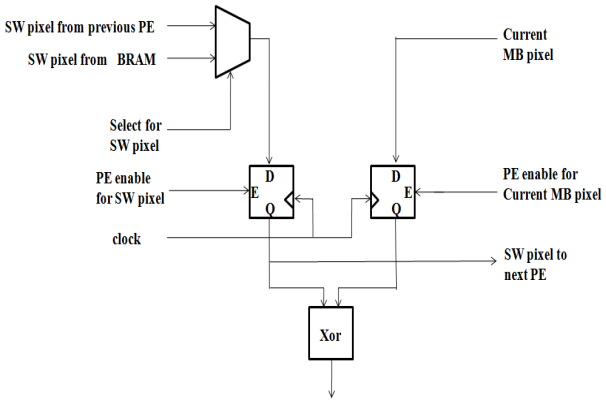
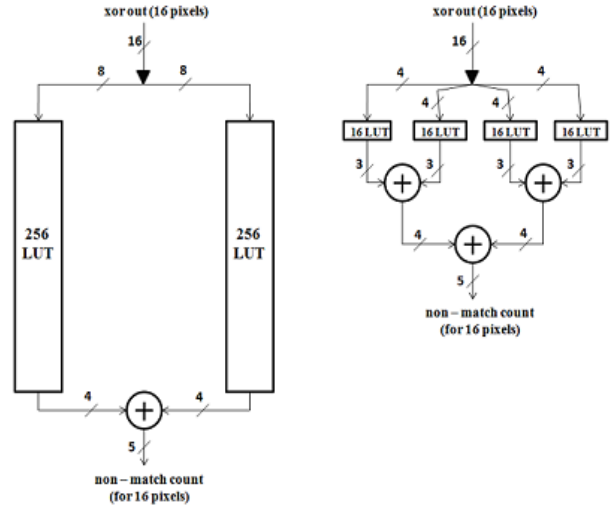


Figure 4. PE Architecture



(a) **(b)**

Figure 5. Non-Match Counter Architectures

(a) Previous Architecture (b) Proposed Architecture

As shown in Figure 5 (a), the Non-Match Counter used in the previous 1BT based ME hardware counts the ones in the outputs of 16 XOR gates by using 2 look up tables with 2^8 entries and adding the outputs of these look up tables. As shown in Figure 5 (b), the Non-Match Counter we propose counts the ones in the outputs of 16 XOR gates by using 4 smaller look up tables with 2^4 entries and adding the outputs of these look up tables. The previous Non-Match Counter consumes 41 slices (82 LUTs) and has a 5.727ns delay. The proposed Non-Match Counter consumes 18 slices (35 LUTs) and has a 3.594ns delay. The proposed Non-Match Counter is faster and has smaller area. Since there are 64 Non-Match Counters in the proposed hardware, the proposed Non-Match Counter provides an area saving of (82-35)*64=3008 LUTs.

B. Memory Organization and Data Alignment

The memory organization of the 1BT based ME hardware architectures proposed in [10] and [15] is shown in Figure 6. These architectures have an inefficient memory organization. They implement full search algorithm for a [-16, 15] search range and a 16x16 MB size. This requires storing a 47x47 pixel = 2209 bits SW in on-chip memory. However, these architectures use 1504x16 = 24064 bits on-chip memory. Because they have pixel duplication in on-chip memory in order to be able to read 2x16 pixels from on-chip memory into PE array in each cycle. As it can be seen in Figure 6, 15 pixels stored in addresses 0 and 47 are the same, and 15 pixels stored in addresses 47 and 94 are the same. Because of this memory organization, the amount of on-chip memory they use for storing a 47x47 pixel SW is more than nine times the on-chip memory needed for storing a 47x47 pixel SW.

This inefficient memory organization also slows down the ME hardware proposed in [10] and [15] because of high loading latency of the on-chip memory. These ME hardware compute the NNMP value for a search location in 1039 clock cycles. If 64 bits can be loaded into on-chip memory from off-chip memory in each clock cycle, 24064 bits on-chip memory for a search location can be loaded in 376 clock cycles and 256 bits current MB can be loaded in 8 clock cycles. This high loading latency of the on-chip memory reduces the performance of these ME hardware.

The memory organization of the 1BT based ME hardware proposed in this paper is shown in Figure 7 and Figure 8. 8 dual-port BRAMs in the FPGA are used to store the 48x96 SW_T. Region 0 includes first 16 lines of the SW_T, Region 1 includes lines 16 to 31 and Region 2 includes lines 32 to 47. Pixels in consecutive two lines of each region are stored in a BRAM. For example, first two lines of each region are stored in BRAM 0, and third and fourth lines of each region are stored in BRAM 1.

18 addresses of each BRAM are used and 32 bits are stored in each address. For example, pixels 0 to 31 in line 0 are stored in address 0 of BRAM 0, pixels 0 to 31 in line 1 are stored in address 1 of BRAM 0 and pixels 0 to 31 in line 2 are stored in address 0 of BRAM 1. Since SW of a single MB requires 2209 bits on-chip memory, SWs of 4 MBs requires 2209x4=8836 bits on-chip memory without data reuse. Because of the efficient data reuse scheme used in the proposed architecture, the proposed architecture uses 8x18x32 = 4608 bits on-chip memory for storing SWs of 4 MBs.

If 64 bits can be loaded into on-chip memory from off-chip memory in each clock cycle, the loading latency of the on-chip memory in the proposed ME hardware is 88 clock cycles; 72 clock cycles for loading 18 addresses of 8 BRAMs and 16 clock cycles for loading current MB pixels into PEs arrays. Therefore on the average 22 clock cycles loading latency is required for one MB which is much smaller than 384 clock cycles loading latency required by previous 1BT based ME hardware architectures.

8 BRAMs provide 16x32 bits in one clock cycle. Therefore, loading the necessary SW pixels for the first search location of a line from BRAMs into a PE array takes 2 clock cycles which is called *line latency*. For the search locations in the first line of SW_T, for all 8 BRAMs, Control Unit generates addresses 0 and 1 in the first clock cycle of *line latency*, addresses 2 and 3 in the second clock cycle of *line latency*, and addresses 4 and 5 in the following 32 clock cycles.

In the first clock cycle of *line latency*, 16x32 bits coming from Vertical Rotator is loaded into MB2 and MB3 PE arrays; the least significant 16 bits of each 32 bits are loaded into MB3 PE array and the most significant 16 bits of each 32 bits are loaded into MB2 PE array. In the second clock cycle of *line latency*, 16x32 bits coming from Vertical Rotator is loaded into MB0 and MB1 PE arrays; the least significant 16 bits of each 32 bits are loaded into MB1 PE array and the most significant 16 bits of each 32 bits are loaded into MB0 PE array. In the following 32 clock cycles after the *line latency*, in each clock cycle, 4 PE arrays compute NNMP values of 4 search locations in the same line.

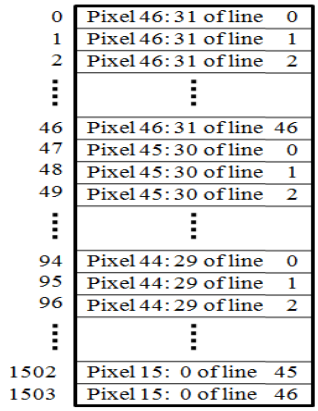


Figure 6. Memory Organization of Previous 1BT based ME Hardware

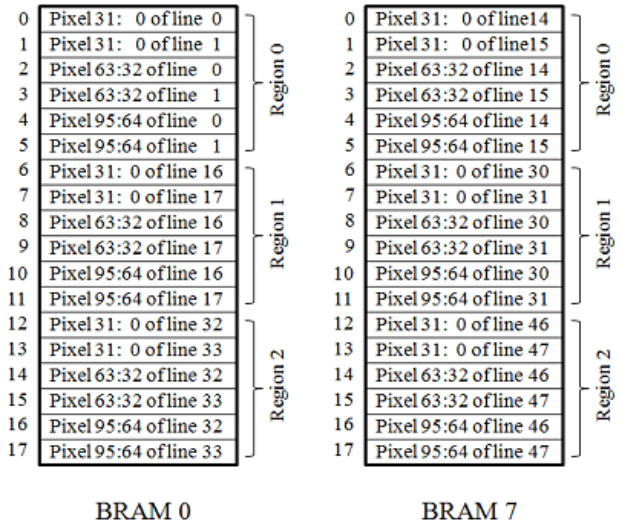


Figure 7. Memory Organization of Proposed 1BT based ME Hardware

Vertical Rotator is used to rotate the SW pixels read from the BRAMs for a search location in order to match them with the corresponding current MB pixels in the PE arrays. Vertical Rotator has 32 identical 16 bit rotators controlled by *rotate amount* signal. Since vertical rotation is not needed for the search locations in the first line of SW_T, *rotate amount* is 0 while computing the NNMP values of these search locations.

For the search locations in the second line of SW_T, in the first clock cycle of *line latency*, Control Unit generates addresses 1 and 6 for BRAM0 and addresses 0 and 1 for the other BRAMs. However, the SW pixels read from the address 1 of BRAM0 should be matched with the current MB pixels in the first row of MB2 and MB3, and the SW pixels read from the address 6 of BRAM0 should be matched with the current MB pixels in the 16^{th} row of MB2 and MB3. Therefore, vertical rotator is used to align the SW pixels with current MB pixels and *rotate amount* signal should be 1. In the second clock cycle of *line latency*, Control Unit generates addresses 3 and 8 for BRAM0 and addresses 2 and 3 for the other BRAMs. In the following 32 clock cycles, Control Unit generates addresses 5 and 10 for BRAM0 and addresses 4 and 5 for the other BRAMs. Therefore, data alignment is needed and *rotate amount* signal should be 1 for all the search locations in line 1 of SW_T.

Since, for the search locations in the lines 16 and 32 of SW_T, SW pixels read from BRAMs and the corresponding current MB pixels in the PE arrays match without vertical rotation, while computing the NNMP values for the search locations in SW_T, *rotate amount* signal takes a value between 0 and 15.

One Bit Selector provides 16 new SW pixels to MB0 PE array for the remaining search locations in a line after the first search location in that line. One Bit Selector is controlled by the *bit select* signal. In the first clock cycle after the NNMP value for the first search location in a line is computed, *bit select* is 0 and the least significant 16 bits coming from vertical rotator are selected and these 16 pixels are sent to MB0 PE array. In the next clock cycle *bit select* is 1 and the second 16 bits coming from vertical rotator are selected and these 16 pixels are sent to MB0 PE array. In this way, *bit select* signal counts from 0 to 31 and in each clock cycle the corresponding 16 new SW pixels are sent to the MB0 PE array. In the last clock cycle *bit select* is 31 and the most significant 16 bits coming from vertical rotator are selected and these 16 pixels are sent to MB0 PE array.

C. Implementation Results

The proposed ME hardware architecture is implemented in Verilog HDL, synthesized with Mentor Graphics Precision RTL 2005b, mapped to a Xilinx XC2VP30-7 FPGA using Xilinx ISE 8.2i. The hardware implementation is verified with post place & route simulation using Mentor Graphics Modelsim 6.1c.

The proposed ME hardware consumes 4758 slices (7280 LUTs), which is 34% of all the slices, of a XC2VP30-7 FPGA. A PE array consumes 547 slices (1094 LUTs), Vertical Rotator consumes 1024 slices (2048 LUTs), One Bit Selector consumes 128 slices (256 LUTs) and the remaining slices are used for Comparator & MV Generator, Control Unit and multiplexers before the address ports of the BRAMs. In addition, 4608 bits on-chip memory is used for storing SWs of 4 MBs, and these 4608 bits are stored in 18 addresses of 8 BRAMs.

Starting the search in a line has a 2 clock cycles *line latency*. Because of the [-16, 16] search range, there are 33 lines in a SW and 33 search locations in each line are searched. 6 stage pipelining causes 6 clock cycles latency. Therefore, ((32+2) x 33) + 6 = 1128 clock cycles are required by the proposed ME hardware for processing 4 MBs and on the average processing one MB requires 282 clock cycles. The proposed ME hardware works at 115 MHz after place & route. Therefore, it is capable of processing 50 1920x1080 full HD frames per second.

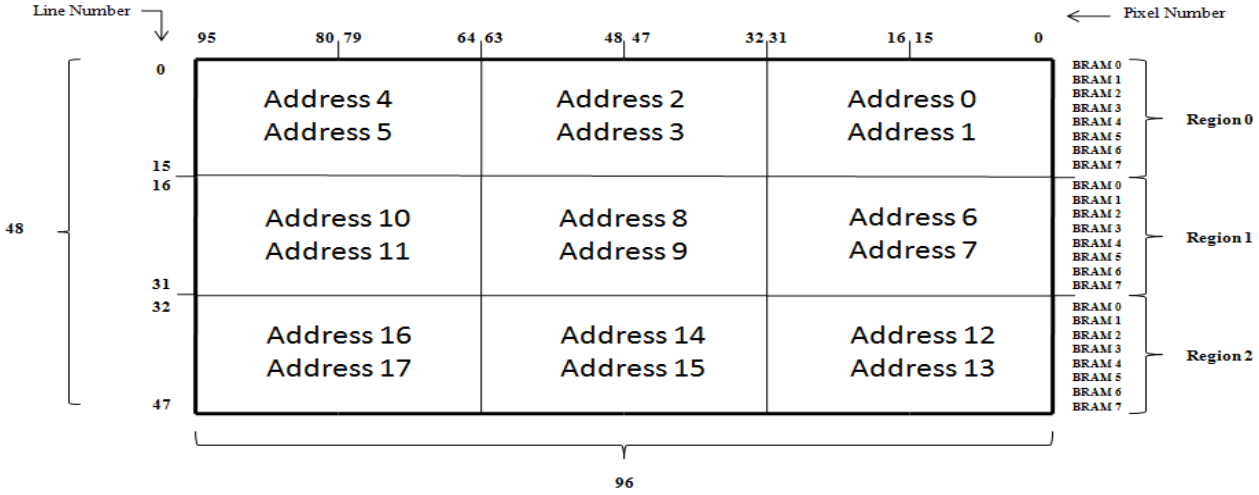


Figure 8. Memory Organization of Proposed 1BT based ME Hardware

Table 1. Comparison of ME Hardware Architectures

	Proposed	[10]	[15]	[8]
Bit Depth	1	1	1	8
On-Chip SW Memory (bits)	4608	24064	24064	30720
LUTs	7280	1589	1467	6648
DFFs	2745	478	499	not reported
FPGA	XC2VP30	XC2VP30	XC2VP30	XC3S1200E
Maximum Frequency (MHz)	115	117	127	144
Search Range	[-16, 16]	[-16,15]	[-16, 15]	[±32, ±16]
Search locations / MB	1089	1024	1024	553
Performance (1920x1080 fps)	50	13	15	25
Performance (1280x720 fps)	113	31	33	57

Comparison of the proposed ME hardware with the ME hardware proposed in [8], [10] and [15] is shown in Table 1. Hardware architectures presented in [10] and [15] are synthesized using Mentor Graphics Precision RTL 2005b and mapped to a Xilinx XC2VP30-7 FPGA using Xilinx ISE 8.2i. The 1BT ME hardware architecture proposed in [10] consumes 998 Slices (1589 LUTs) and 24064 bits on-chip memory for storing the search window. It requires 1039 clock cycles for processing one MB for a [-16, 15] search range. It works at 117 MHz and it can process 13 1920x1080 full HD frames per second. The MF1BT ME hardware architecture proposed in [15] consumes 944 Slices (1467 LUTs) and 24064 bits on-chip memory for storing the search window. It requires 1039 clock cycles for processing one MB for a [-16, 15] search range. It works at 127 MHz and it can process 15 1920x1080 full HD frames per second.

The area of the proposed 1BT based ME hardware is larger than the area of the 1BT based ME hardware proposed in [10] and [15] because of performing ME for 4 MBs in parallel and data alignment. However, the proposed ME hardware is much faster and has much less on-chip memory than these ME hardware.

The 1BT based ME hardware architecture proposed in this paper is based on the ME hardware architecture proposed in [8]. However, the ME hardware proposed in [8] is performing ME for 8-bit pixels, using SAD block matching criterion, and it is not performing ME for 4 MBs in parallel. The ME hardware proposed in [8] is implemented on a XC3S1200E-5 FPGA. It can process 25 1920x1080 full HD frames per second for a [±32, ±16] search range. It consumes 6648 LUTs and 30720 bits on-chip memory for storing the search window. The 1BT based ME hardware proposed in this paper has nearly same logic area with this ME hardware, but it has higher performance and much less on-chip memory.

4. CONCLUSION

In this paper, we presented a high performance systolic hardware architecture for 1BT based ME. The proposed hardware performs full search ME for 4 MBs in parallel and it is the fastest 1BT based ME hardware reported in the literature. In addition, it uses less on-chip memory than the previous 1BT based ME hardware by using a novel data reuse scheme and memory organization. The proposed hardware consumes %34 of the slices in a Xilinx XC2VP30-7 FPGA. It works at 115 MHz in the same FPGA and is capable of processing 50 1920x1080 full HD frames per second. Therefore, it can be used in consumer electronics products that require real-time video processing or compression.

ACKNOWLEDGEMENTS

This research was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) under the contract 107E179.

REFERENCES

[1] I. Richardson, *H.264 and MPEG-4 Video Compression,* Wiley, 2003.

[2] B.-D. Choi, J.-W. Han, C.-S. Kim, S.-J. Ko, "Motion-compensated Frame Interpolation Using Bilateral Motion Estimation and Adaptive Overlapped Block Motion Compensation," *IEEE Trans. on Circuits Syst. Video Technol.,* vol. 17, no. 4, pp. 407–416, Apr. 2007.

[3] Y. Ling, J. Wang, Y. Liu, and W. Zhang, "A Novel Spatial and Temporal Correlation Integrated Based Motion-compensated Interpolation for Frame Rate Up-conversion," *IEEE Trans. on Consumer Electronics*, vol. 54, no. 2, pp. 863-869, May 2008.

[4] R. Li, B. Zeng, and M.L. Liou, "A New Three-step Search Algorithm for Block Motion Estimation,"

IEEE Trans. on Circuits Syst. Video Technol., vol. 4, pp. 438–442, 1994.

[5] S. Zhu and K.-K. Ma, "A New Diamond Search Algorithm for Fast Block Matching Motion Estimation," *IEEE Trans. on Image Processing*, vol. 9, pp. 287–290, 2000.

[6] C. Zhu, X. Lin, and L. P. Chau, "Hexagon-based Search Pattern for Fast Block Motion Estimation," *IEEE Trans. on Circuits Syst. Video Technol.,* vol. 12, pp. 349–355, 2002.

[7] X.-Q. Banh and Y.-P. Tan, "Adaptive Dual-cross Search Algorithm for Block-matching Motion Estimation", *IEEE Trans. on Consumer Electronics*, vol. 50, no. 2, pp. 766-775, May 2004.

[8] O. Tasdizen, A. Akin, H. Kukner, I. Hamzaoglu, and H. F. Ugurdag, "High Performance Hardware Architectures for a Hexagon-based Motion Estimation Algorithm," *16th IEEE / IFIP International Conference on VLSI - SoC*, Rhodes Island, Greece, Oct. 2008.

[9] O. Tasdizen, H. Kukner, A. Akin and I. Hamzaoglu, " A High Performance Reconfigurable Motion Estimation Hardware Architecture," *IEEE DATE Conference*, Nice, France, Apr. 2009.

[10] B. Natarajan, V. Bhaskaran, K. Konstantinides, "Low-Complexity Block-based Motion Estimation via One-bit Transforms," *IEEE Trans. on Circuits Syst. Video Technol.,* vol. 7, no. 3, pp. 702-706, Aug. 1997.

[11] S. Ertürk, "Multiplication-Free One-Bit Transform for Low-Complexity Block-Based Motion Estimation," *IEEE Signal Process. Lett.*, vol. 14, no. 2, pp. 109-112, Feb. 2007.

[12] H. Lee and J. Jeong, "Early Termination Scheme for Binary Block Motion Estimation," *IEEE Trans. on Consumer Electronics*, vol. 53, no. 4, pp. 1682-1686, Nov. 2007.

[13] A. Ertürk, S. Ertürk, "Two-Bit Transform for Binary Block Motion Estimation," *IEEE Trans. on Circuits Syst. Video Technol.,* vol. 15, no. 7, pp. 938-946, July 2005.

[14] O. Urhan and S. Ertürk, "Constrained One-Bit Transform for Low Complexity Block Motion Estimation," *IEEE Trans. on Circuits Syst. Video Technol.,* vol. 17, no. 4, pp. 478-482, Apr. 2007.

[15] A. Çelebi, O. Urhan, İ. Hamzaoğlu, S. Ertürk, "Efficient Hardware Implementations of Low Bit Depth Motion Estimation Algorithms," *IEEE Signal Process. Letts.*, vol. 16, no. 6, June 2009.

[16] A. Çelebi, O. Akbulut, O. Urhan, İ. Hamzaoğlu, S. Ertürk, "An All Binary Sub-Pixel Motion Estimation Approach and its Hardware Architecture," *IEEE Trans. on Consumer Electronics*, vol. 54, no. 4, Nov. 2008.

[17] J.-C. Tuan, T.-S. Chang, and C.-W. Jen, "On the Data Reuse and Memory Bandwidth Analysis for Full-search Block-matching VLSI Architecture", *IEEE Trans. on Circuits Syst. Video Technol., v*ol. 12, pp. 61-72, 2002.

[18] S. Goel, Y. Ismail, P. Devulapalli, J. McNeely, M. Bayoumi, "An Efficient Data Reuse Motion Estimation Engine", *IEEE Signal Process. Syst. Design and Imp.*, pp. 383-386, Oct. 2006.