HIGH PERFORMANCE HARDWARE ARCHITECTURES FOR ONE BIT TRANSFORM
BASED MOTION ESTIMATION

by
ABDULKADİR AKIN

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University
Spring 2010

I

HIGH PERFORMANCE HARDWARE ARCHITECTURES FOR ONE BIT TRANSFORM BASED MOTION ESTIMATION

APPROVED BY:

Assist. Prof. Dr. İlker Hamzaoğlu   ………………………….

(Thesis Supervisor)

Assoc. Prof. Dr. Erkay Savaş   ………………………….

Assist. Prof. Dr. Hakan Erdoğan   ………………………….

Assoc. Prof. Dr. Oğuzhan Urhan   ………………………….

Prof. Dr. Sarp Erturk   ………………………….

DATE OF APPROVAL: ………………………….

HIGH PERFORMANCE HARDWARE ARCHITECTURES FOR ONE BIT TRANSFORM BASED MOTION ESTIMATION

Abdulkadir Akın

EE, Master Thesis, 2010

Thesis Supervisor: Assist. Prof. Dr. İlker Hamzaoğlu

ABSTRACT

Motion Estimation (ME) is the most computationally intensive and most power consuming part of video compression and video enhancement systems. ME is used in video compression standards such as MPEG4, H.264 and it is used in video enhancement algorithms such as frame rate conversion and de-interlacing.

One bit transform (1BT) based ME algorithms have low computational complexity. Therefore, in this thesis, we propose high performance hardware architectures for 1BT based fixed block size (FBS) single reference frame (SRF) ME, variable block size (VBS) SRF ME, and multiple reference frame (MRF) ME. Constraint One Bit Transform (C-1BT) ME algorithm improves the ME performance of 1BT ME, and the early terminated C-1BT ME algorithm reduces the computational complexity of C-1BT ME. Therefore, in this thesis, we also propose a high performance early terminated C-1BT ME hardware architecture.

The proposed FBS SRF ME hardware architectures perform full search ME for 4 Macroblocks in parallel and they are faster than the 1BT based ME hardware reported in the literature. In addition, they use less on-chip memory than the previous 1BT based ME hardware by using a novel data reuse scheme and memory organization. The proposed VBS SRF ME and MRF ME hardware architectures are the first 1BT based VBS ME and MRF ME hardware architectures in the literature. The proposed MRF ME hardware is designed as reconfigurable in order to statically configure the number and selection of reference frames based on the application requirements. The proposed early terminated C-1BT ME hardware architecture is the first early terminated C-1BT ME hardware architecture in the literature.

All of the proposed ME hardware architectures are implemented in Verilog HDL and mapped to Xilinx FPGAs. All FPGA implementations are verified with post place & route simulations.

1 BİT DÖNÜŞÜMÜ TEMELLİ HAREKET TAHMİNİ ALGORİTMALARI İÇİN YÜKSEK PERFORMANSLI DONANIM MİMARİLERİ

Abdulkadir Akın

EE, Yüksek Lisans Tezi, 2010

Tez Danışmanı: Yard. Doç. Dr. İlker Hamzaoğlu

ÖZET

Hareket Tahmini (HT) video sıkıştırma ve video iyileştirme sistemlerinin en çok işlem yapan ve en çok güç harcayan kısmıdır. HT, MPEG4 ve H.264 gibi video sıkıştırma standartlarında ve çerçeve hızı dönüştürme gibi video iyileştirme işlerinde kullanılır.

1 Bit Dönüşümü (1BD) temelli HT algoritmalarının işlemsel karmaşıklığı düşüktür. Bu nedenle, bu tezde yüksek performanslı 1BD temelli sabit blok boyutlu (SBB) tek referans çerçeve (TRÇ) HT, değişken blok boyutlu (DBB) TRÇ HT ve SBB çoklu referans çerçeve (ÇRÇ) HT donanım mimarileri önerdik. Önerilen SBB TRÇ HT donanımları 4 makroblok için HT işlemlerini paralel olarak yapmaktadır ve tam arama algoritmasını kullanmaktadır. Önerilen SBB TRÇ HT donanımları literatürdeki 1BD temelli HT donanımlarından daha hızlıdır. Önerilen donanımlar verileri tekrar kullanma yöntemleri ve etkili bellek organizasyonları kullandıkları için literatürdeki 1BD temelli HT donanımlarından daha az yonga-üzeri-bellek kullanmaktadırlar. Önerilen ÇRÇ HT donanımı ve önerilen DBB TRÇ HT donanımı literatürdeki ilk 1BD temelli ÇRÇ HT ve DBB TRÇ HT donanımlarıdır. ÇRÇ HT donanımı yeniden yapılandırılabilir şekilde tasarlanmıştır. Statik olarak yeniden yapılandırılabilme özelliği sayesinde HT yapılacak uygulamanın gerekesinimlerine göre arama işlemi yapılacak referans çerçevelerin sayısı ve seçimi yapılabilmektedir.

Kısıtlanmış 1BD (K-1BD) HT algoritması 1BD HT algoritmasının HT performansını arttırmaktadır. Erken sonlandırma yöntemi kullanan K-1BD HT algoritması ise K-1BD HT algoritmasının işlemsel karmaşıklığını azaltmaktadır. Bu nedenle, bu tezde erken sonlandırma yöntemini kullanan K-1BD HT algoritmasını gerçekleyen bir donanım tasarladık.

Bu tezde önerilen bütün donanım mimarileri Verilog HDL ile gerçeklendiler ve benzetimleri yapılarak doğrulandılar.

To my sister Elif…

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

1BT   One Bit Transform

2BT   Two Bit Transform

ASIC   Application Specific Integrated Circuit

BM   Block Matching

BRAM   Block Ram

C-1BT   Constraint One Bit Transform

CM   Constraint Mask

CNNMP   Constraint Number of Non-Matching Pixels

DFF   D Flip-Flop

FBS   Fixed Block Size

FPGA   Field Programmable Gate Array

FS   Full Search

HD   High Definition

HDL   Hardware Description Language

Hz   Hertz

LUT   Look Up Table

MB   Macroblock

ME   Motion Estimation

MF1BT   Multiplication Free One Bit Transform

MRF   Multiple Reference Frame

MV   Motion Vector

NNMP   Number of Non-Matching Pixels

NMC   Non-Match Counter

PE   Processing Element

PSNR   Peak Signal to Noise Ratio

RF   Reference Frame

RTL   Register Transfer Level

SAD   Sum of Absolute Differences

SRF   Single Reference Frame

SW   Search Window

SW$_T$ Total Search Window

VBS   Variable Block Size

CHAPTER I

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 the block from the 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.

Among the BM algorithms, Full Search (FS) algorithm achieves the best performance since it searches all search locations in a given search range. But the computational complexity of FS ME algorithm is high. In order to improve the ME performance, variable block size (VBS) and multiple reference frame (MRF) ME are used in H.264 standard. But the computational complexity of FS algorithm for VBS ME and MRF ME is even higher [4, 5, 6, 7].

Several fast search ME algorithms, such as New Three Step Search [8], Diamond Search [9], Hexagon-Based Search [10], and Adaptive Dual Cross Search [11], 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 [12, 13].

1

Another preferred method for reducing the computational complexity of FS algorithm is reducing pixel resolution from 8 bits to fewer bits. In [14], 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 [14], 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 [15], 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 [16]. In [17], two bit transform (2BT) is proposed to improve ME accuracy compared to 1BT by constructing two bit-planes for each frame and performing ME using 2BT representations. In [18], constraint one-bit transform (C-1BT) is proposed and it is shown that C-1BT provides increased ME accuracy compared to 2BT at a lower complexity.

The first 1BT based ME hardware implementation in the literature is presented in [14]. In [14], a motion vector (MV) based linear arrays hardware architecture is used for implementing 1BT based ME. In [19], a source pixel based linear arrays hardware architecture is used for implementing low bit depth ME algorithms proposed in [14] and [15]. In [20], a new sub-pixel accurate low bit depth ME algorithm and its hardware is presented. In [21], low bit depth motion estimation hardware is used in an H.264 encoder for mobile applications.

In this thesis, we propose high performance hardware architectures for 1BT based ME algorithms. The proposed ME hardware architectures perform full search ME for 4 Macroblocks (MBs) in parallel and they are faster than the 1BT based ME hardware reported in the literature. They use less on-chip memory than the previous 1BT based ME hardware by using a novel data reuse scheme and memory organization.

First, we propose high performance systolic hardware architectures for 1BT based FBS ME and VBS ME with rectangular MB organization [22, 23]. The proposed 1BT ME

hardware architectures are based on the 8 bits/pixel FBS ME hardware architecture proposed in [13]. The major differences between them are the proposed ME hardware architectures calculate MVs of 4 MBs in parallel, use a novel data reuse scheme, and use less on-chip memory, processing element array and adder tree area because of 1BT. Data reuse method is used for reducing the off-chip and on-chip memory bandwidth required by ME hardware.

The proposed ME hardware architectures using rectangular MB organization are faster than the 1BT based ME hardware architectures reported in the literature and they are capable of processing 1920x1080 full High Definition (HD) videos in real-time. The 1BT based ME hardware proposed in [14, 19] 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 [14, 19]. Although the proposed ME hardware store search windows of 4 MBs in on-chip memory, they use less on-chip memory and they load the on-chip memory from off-chip memory in less number of clock cycles than the ME hardware proposed in [14, 19].

The proposed FBS ME and VBS ME hardware architectures using rectangular MB organization are implemented in Verilog HDL. The Verilog RTL codes are verified by simulation using Mentor Graphics Modelsim. They are mapped to Xilinx *XC2VP30-7* FPGA using Xilinx ISE. FBS ME hardware consumes 4758 slices (34% of all the slices) and 8 BlockRAMs (BRAMs). VBS ME hardware consumes 6782 slices (49% of all the slices) and 8 BRAMs. Both FBS ME and VBS ME hardware can work at 113 MHz, and they are capable of processing 49 1920x1080 full HD frames per second.

Then, we propose high performance hardware architectures for 1BT based FBS SRF ME, VBS SRF ME and FBS MRF ME with square MB organization [24, 25]. Both the proposed FBS SRF ME hardware using square MB organization and the proposed FBS SRF ME hardware using rectangular MB organization search 4 MBs in parallel. However, their data reuse schemes, memory organizations, PE arrays and data alignment schemes are different. The proposed FBS SRF ME hardware using square MB organization is faster, uses less on-chip memory, and loads the on-chip memory in less number of clock cycles than the FBS SRF ME hardware reported in [14], [19]. The proposed FBS SRF ME hardware using square MB organization is faster, uses less on-chip memory, uses less logic area and loads the on-chip memory in less number of clock cycles than the proposed FBS SRF ME hardware

3

using rectangular MB organization. The proposed VBS SRF ME hardware using square MB organization is also faster, and uses less logic area and on-chip memory than the 1BT based VBS SRF ME hardware using rectangular MB organization.

In addition, we propose a high performance reconfigurable hardware architecture for 1BT based FBS MRF ME [25]. This is the first 1BT based MRF ME hardware in the literature. In the proposed reconfigurable MRF ME hardware, the number and selection of reference frames can be statically configured based on the application requirements in order to trade-off ME performance and computational complexity.

The proposed 1BT based FBS SRF ME, VBS SRF ME and MRF ME hardware architectures using square MB organization are implemented in Verilog HDL and mapped to Xilinx *XC2VP30-7* FPGA using Xilinx ISE. They are all capable of processing 83 1920x1080 full HD frames per second.

Finally, we propose high performance systolic hardware architectures for C-1BT ME and early terminated C-1BT ME. The proposed C-1BT ME and early terminated C-1BT ME hardware architectures are implemented in Verilog HDL and mapped to Xilinx *XC2VP30-7* FPGA using Xilinx ISE. They are all capable of processing 83 1920x1080 full HD frames per second. The power consumptions of both ME hardware on Virtex 5 FPGA are estimated using Xilinx XPower Analyzer. Based on the power estimation results, early terminated C-1BT ME hardware consumes 17% less energy than C-1BT ME hardware for a full HD frame in which 40% of the MBs are early terminated.

The rest of the thesis is organized as follows;

Chapter II explains 1BT based ME algorithms.

Chapter III presents the proposed high performance 1BT based FBS SRF ME and VBS SRF ME hardware architectures with rectangular MB organization.

Chapter IV presents the proposed high performance 1BT based FBS SRF ME, VBS SRF ME and MRF ME hardware architectures with square MB organization. In this Chapter, the simulation results for 1BT based SRF ME and MRF ME algorithms are also presented.

Chapter V presents the proposed high performance hardware architectures for C-1BT ME and early terminated C-1BT ME algorithms.

Chapter VI presents the conclusions and the future work.

CHAPTER II

ONE BIT TRANSFORM BASED MOTION ESTIMATION ALGORITHMS

Motion estimation is the process of searching a search window in a reference frame to determine the best match for a block in a current frame based on a search criterion such as minimum Sum of Absolute Difference [1]. As shown in Figure 2.1, the location of a block in a frame is given using the (x,y) coordinates of top-left corner of the block. The search window in the reference frame is the [-p,p] size region around the location of the current block in the current frame. The SAD value for a current block in the current frame and a candidate block in the reference frame is calculated by accumulating the absolute differences of corresponding pixels in the two blocks as shown in the formula (2.1).

$$(2.1)$$

In formula (2.1), B_{mxn} is a block of size mxn, $\mathbf{d}=(dx,\ dy)$ is the motion vector, *c* and *r* are current and reference frames respectively. Since a motion vector expresses the relative motion of the current block in the reference frame, motion vectors are specified in relative coordinates. If the location of the best matching block in the reference frame is (x+u, y+v), then the motion vector is expressed as (u,v). Motion estimation is performed on the luminance (Y) component of a YUV image and the resulting motion vectors are also used for the chrominance (U and V) components.
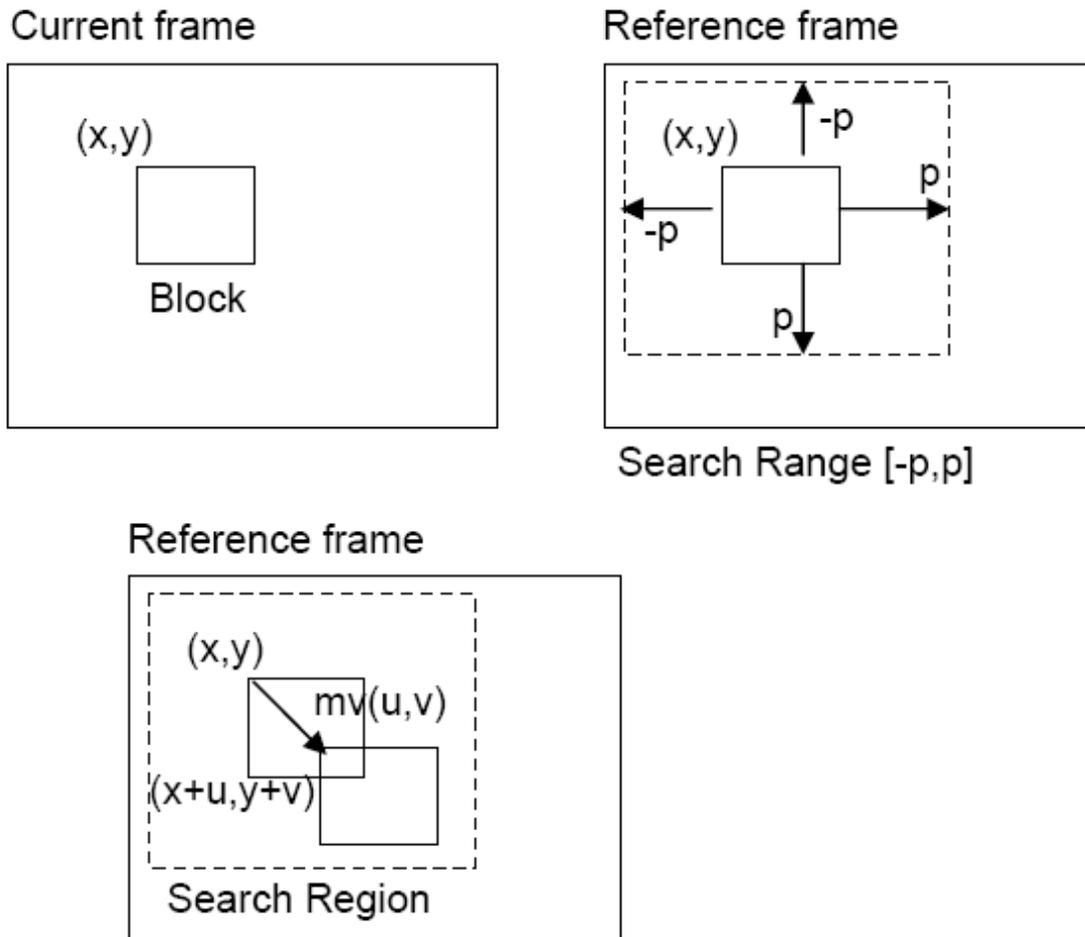
Figure 2.1 Motion Estimation

Full Search ME algorithm finds the reference block that best matches the current block by computing the SAD values for all search locations in a given search range. Although many fast search ME algorithms are developed, FS algorithm has remained a popular candidate for hardware implementation because of its regular dataflow and good compression performance [26, 27]. Since FS algorithm has a high computational complexity, FS ME hardware consume large amount of power, logic area and on-chip memory.

In order to reduce computational complexity of 8 bit depth FS ME, 1BT ME algorithm is proposed in [14]. In [14], 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 [15], a novel diamond shape kernel (2.2) is proposed to decrease the computational complexity of the

filtering stage of 1BT. This new kernel contains 16 non-zero elements and thus the multiplication operation becomes simple logical shift. 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 (2.3). The number of non-matching points (NNMP) criterion proposed in [14] is then used to evaluate the match of two blocks as shown in (2.4). The symbol $\oplus$ denotes XOR operation. The search location which has the smallest *NNMP* value is selected as the MV of the current block. Figure 2.2 shows a sample image from coastguard video sequence, its filtered version, corresponding one bit depth image and the reconstructed image.

$$K = \frac{1}{16}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.2)$$

$$B(i, j) = \begin{cases} 1, & \text{if } I(i, j) \geq I_F(i, j) \\ 0, & \text{otherwise} \end{cases} \quad (2.3)$$

$$NNMP\ (m, n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} B^t(i, j) \oplus B^{t-1}(i + m, j + n) \quad (2.4)$$

8
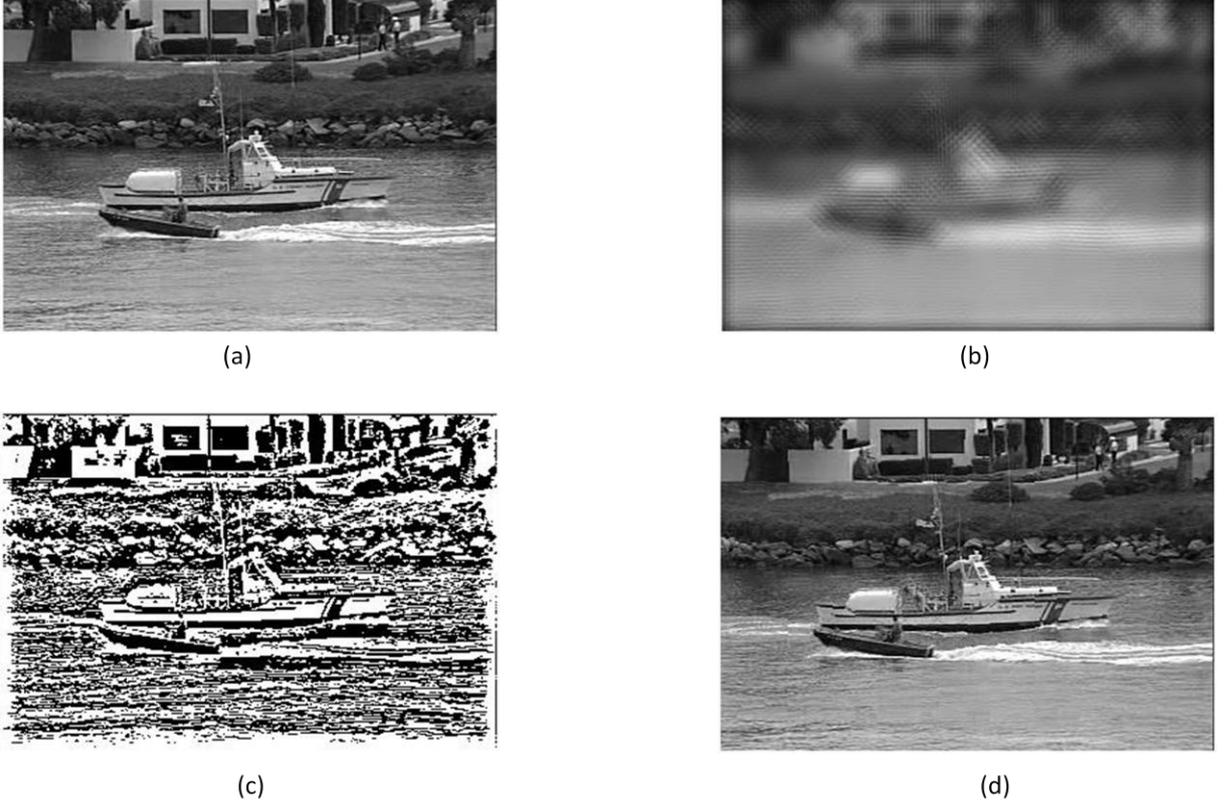
Figure 2.2 (a) Original Image **(b)** Filtered Image **(c)** One Bit Depth Image **(d)** Reconstructed Image

In [18], Constraint 1BT (C-1BT) motion estimation algorithm is proposed to improve the motion estimation performance of MF1BT. In C-1BT, a constraint mask (CM) is constructed. As shown in (2.5), CM value of a pixel is 1 if the pixel is more than a certain distance (D) away from the transform threshold. Constraint NNMP (CNNMP) criterion uses CM to decide whether pixels can be reliably used for 1BT matching or not. As shown in (2.6), if at least one of the two pixels has a CM value of 1, 1BT matching is used to determine whether these two pixels match or not. If both pixels have a CM value of 0, 1BT matching is not used for these pixels and they are not counted as a non-match.

$$CM(i,j) = \begin{cases} 1, & if \left| I(i,j) - I_F(i,j) \right| \geq D \\ 0, & otherwise \end{cases} \tag{2.5}$$

$$CNNMP(m,n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \overline{\left[CM^t(i,j) \| CM^{t-1}(i+m,j+n)\right]} \ \& \ \overline{\left[B^t(i,j) \oplus B^{t-1}(i+m,j+n)\right]} \tag{2.6}$$

In order to reduce the computational complexity of C-1BT algorithm, early terminated C-1BT algorithm is proposed in [28]. As shown in (2.7), if the number of "1"s in CM of a block is less than a threshold, early terminated C-1BT algorithm decides that the block is stable and computes its motion vector by taking the median of the motion vectors of the upper, left and upper-left neighboring blocks. The variable   is used to determine the threshold value for a frame, and it is calculated using the formula (2.8) in which A is set to -16 and B is set to 12 experimentally.

$$\overline{} \qquad\qquad (2.7)$$

$$\overline{} \qquad\qquad (2.8)$$

CHAPTER III

HIGH PERFORMANCE HARDWARE ARCHITECTURES FOR ONE BIT TRANSFORM BASED MOTION ESTIMATION WITH RECTANGULAR MACROBLOCK ORGANIZATION

3.1 Proposed Hardware Architecture for One Bit Transform based Fixed Block Size Single Reference Frame Motion Estimation

The block diagram of the proposed hardware for 1BT based FBS ME is shown in Figure 3.1. 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 Fig. 3.2. 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).

3.1.1 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 3.3. 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.
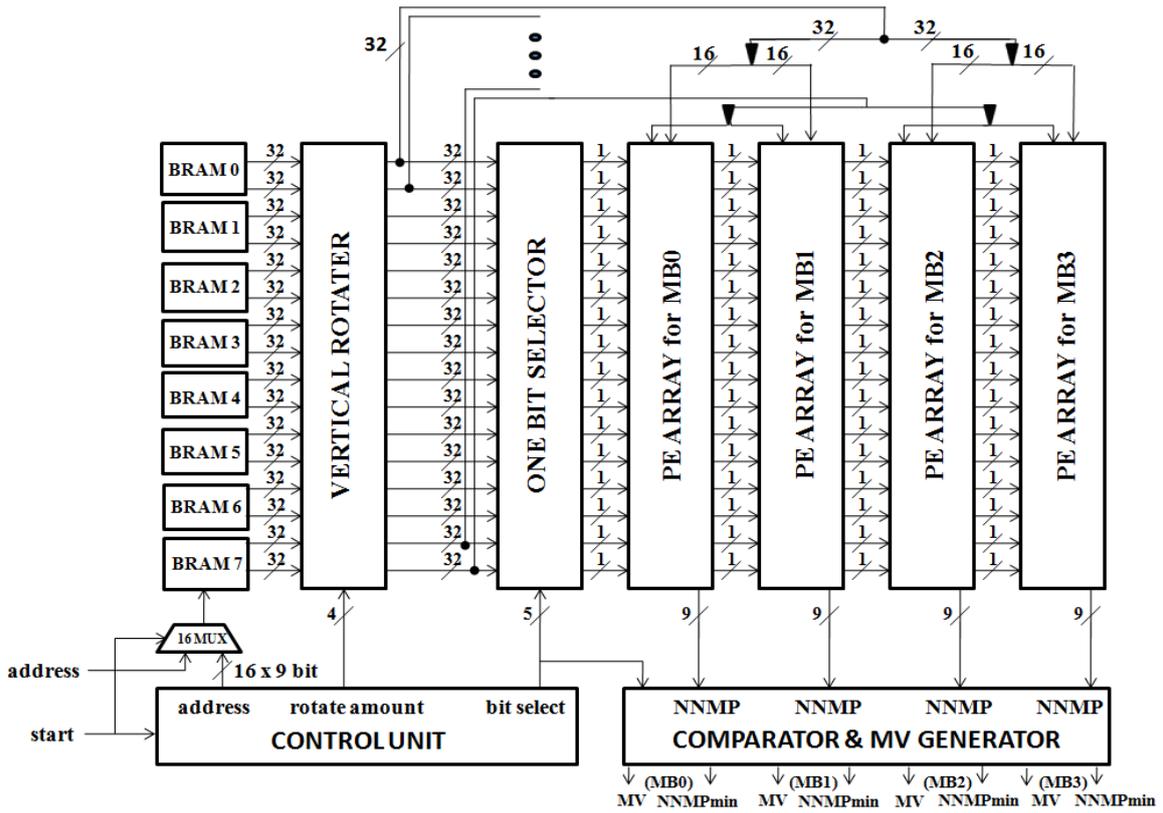
Figure 3.1 Top-level Block Diagram of Proposed FBS ME Hardware
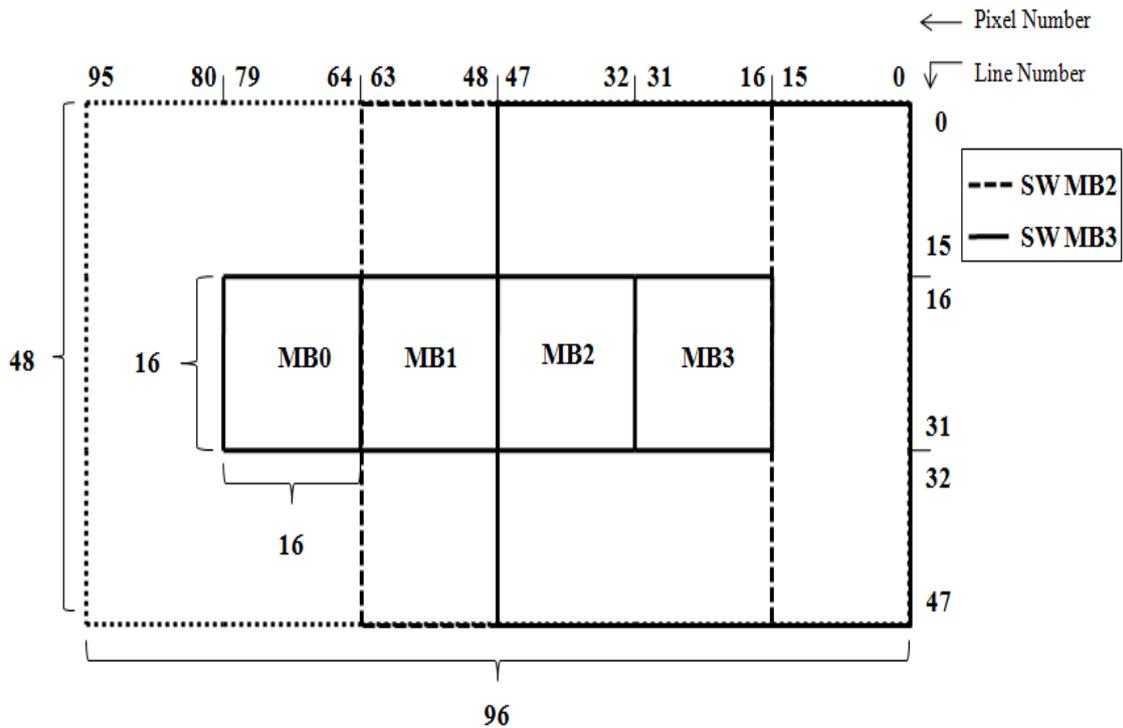
Figure 3.2 Search Windows of 4 Macroblocks
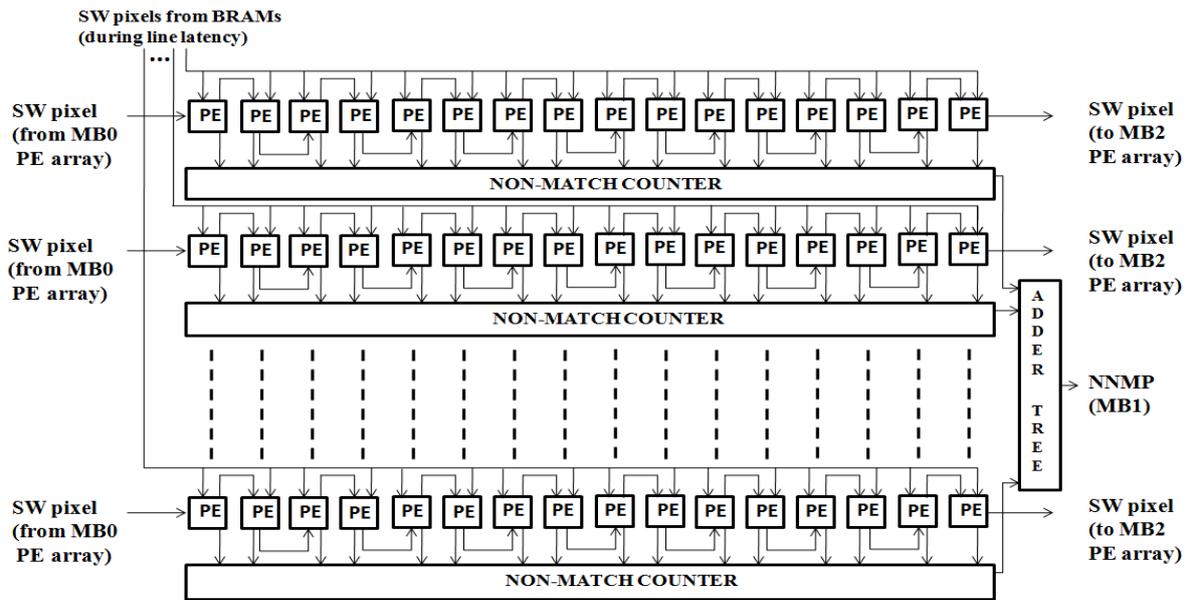
Figure 3.3 MB1 PE Array for FBS ME Hardware
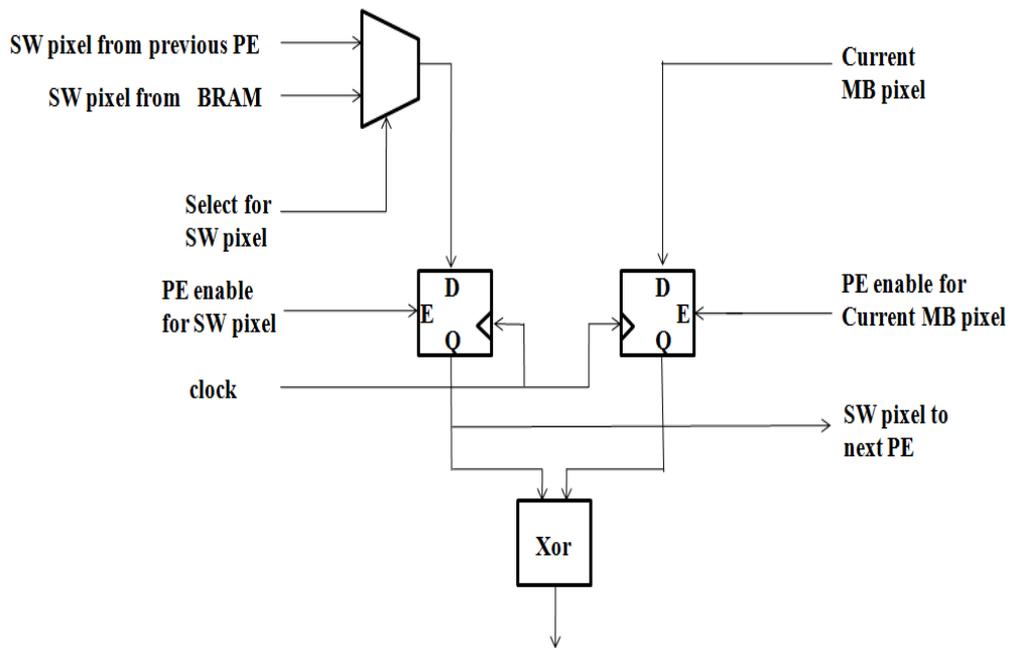
Figure 3.4 PE Architecture
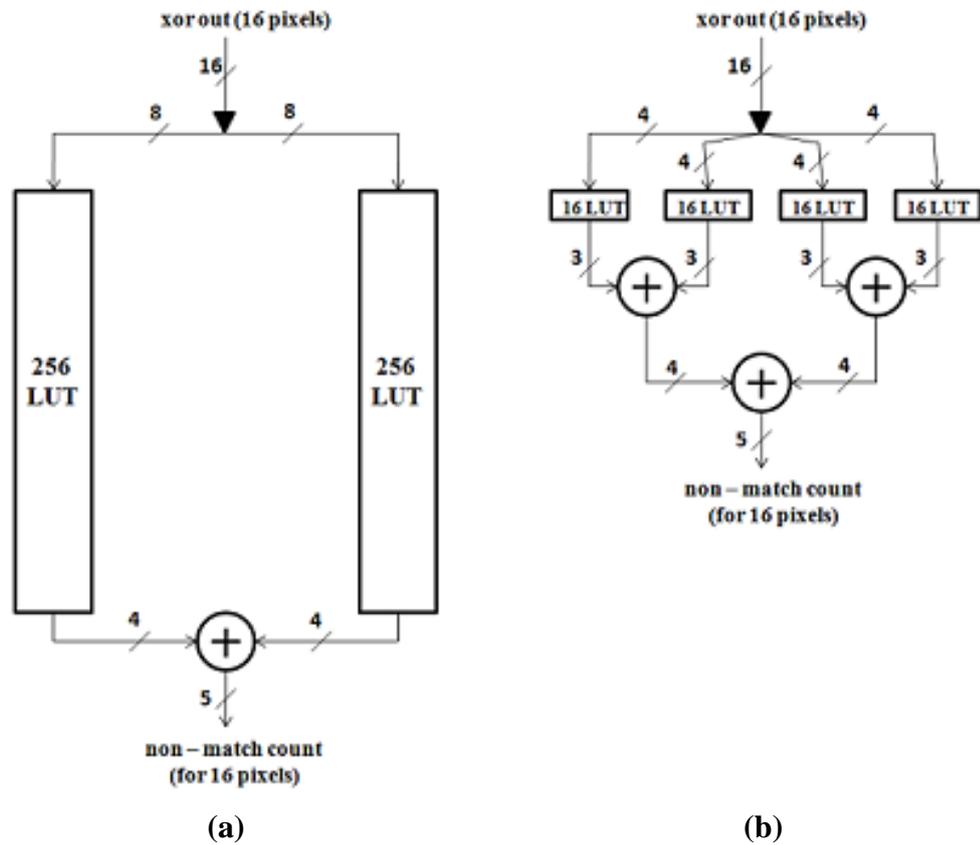
Figure 3.5 Non-Match Counter Architectures **(a)** Previous Architecture **(b)** Proposed
Architecture

The architecture of a PE is shown in Figure 3.4. 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.

As shown in Figure 3.5 (a), the Non-Match Counter used in the previous 1BT based ME hardware architectures in literature 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 3.5 (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.

3.1.2 Memory Organization and Data Alignment

The memory organization of the 1BT based ME hardware architectures proposed in [14, 19] is shown in Figure 3.6. 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 3.6, 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 [14, 19] 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 thesis is shown in Figure 3.7 and Figure 3.8. 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 require 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.

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 16th 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$_T$.

0	Pixel 46:31 of line 0
1	Pixel 46:31 of line 1
2	Pixel 46:31 of line 2
⋮	⋮
46	Pixel 46:31 of line 46
47	Pixel 45:30 of line 0
48	Pixel 45:30 of line 1
49	Pixel 45:30 of line 2
⋮	⋮
94	Pixel 44:29 of line 0
95	Pixel 44:29 of line 1
96	Pixel 44:29 of line 2
⋮	⋮
1502	Pixel 15: 0 of line 45
1503	Pixel 15: 0 of line 46

Figure 3.6 Memory Organization of Previous 1BT based ME Hardware

Figure 3.7 Memory Organization of Proposed 1BT based ME Hardware

Figure 3.8 Memory Organization of Proposed 1BT based ME Hardware

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

19

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.

3.2 Proposed Hardware Architecture for One Bit Transform Based Variable Block Size Motion Estimation

The top-level block diagram of the proposed VBS ME hardware architecture is similar to the top-level block diagram of the proposed FBS ME hardware architecture shown in Figure 3.1. The Non-Match Counters (NMC) and Adder Trees used in the PE arrays in FBS ME hardware and the ones used in the PE arrays in VBS ME hardware are different. MB1 PE array for VBS ME hardware is shown in Figure 3.9. As shown in Figure 3.3 and Figure 3.9, even though each PE array in FBS ME hardware computes the NNMP value for a MB, each PE array in VBS ME hardware computes the NNMP values for the 41 partitions of a MB. The 41 partitions of a MB are shown in Figure 3.10.

In VBS ME hardware, a NMC in a PE array computes the NNMP value for 4 current MB and 4 SW pixels using a look up table with 2^4 entries. In the Adder Tree, the outputs of the NMCs are added to compute the NNMP values for the 16 4x4 blocks. For example, NMC (0, 0), NMC (1, 0), NMC (2, 0) and NMC (3, 0) are added to compute the NNMP value for 4x4 block 1 as shown in Figure 3.10. The NNMP values for the 4x4 blocks are added to

compute the NNMP values for the 4x8 and 8x4 blocks and these NNMP values are stored in pipeline registers.

The NNMP values for the 4x8 blocks are added to compute the NNMP values for the 8x8 blocks. The NNMP values for the 8x8 blocks are added to compute the NNMP values for the 8x16 and 16x8 blocks and these NNMP values are stored in pipeline registers. The NNMP values for the 8x16 blocks are added to compute the NNMP value for the 16x16 block. Therefore, the pipelining in the Adder Trees in VBS ME hardware causes 2 clock cycles latency for computing the NNMP value for a 16x16 MB same as the Adder Trees in FBS ME hardware.

The 41 NNMP values of each MB computed by a PE array are sent to Comparator & MV Generator, and the Comparator & MV Generator determines the minimum NNMP values and the corresponding MVs for each MB partition.
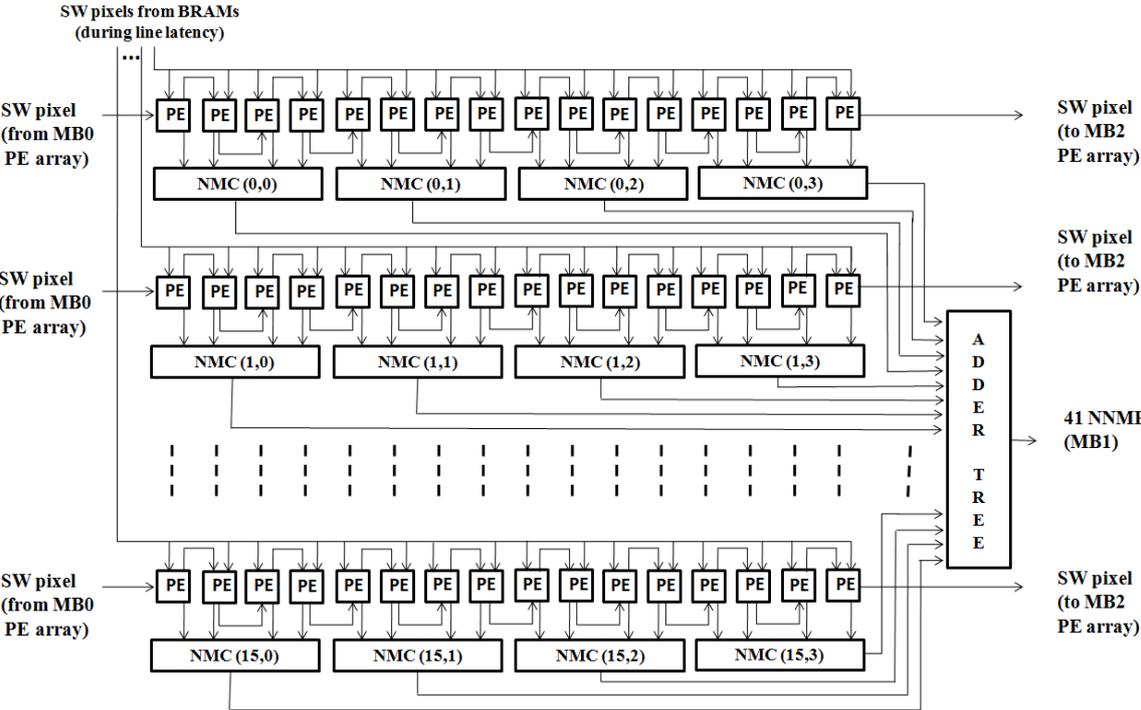
Figure 3.9 MB1 PE Array for VBS ME Hardware

21
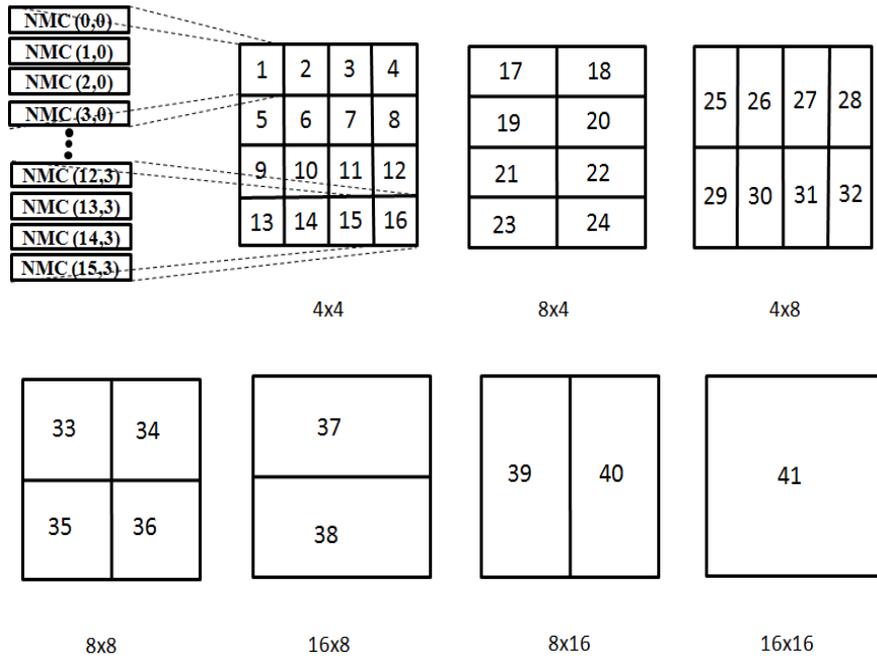
Figure 3.10 Macroblock Partitions

3.3 Implementation Results

The proposed 1BT based FBS ME and VBS ME hardware architectures are implemented in Verilog HDL. The Verilog RTL codes are synthesized with Mentor Graphics Precision RTL 2005b and mapped to a Xilinx XC2VP30-7 FPGA using Xilinx ISE 8.2i. The hardware implementations are verified with post place & route simulations using Mentor Graphics Modelsim 6.1c.

The FBS 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 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.

The VBS ME hardware consumes 6782 slices (8702 LUTs) which is 49% of all the slices of a XC2VP30-7 FPGA. Since VBS ME hardware has more complex Comparator & MV Generator and Adder Tree than FBS ME hardware, it consumes more LUTs and DFFs than FBS ME hardware.

For both FBS ME and VBS ME hardware, 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 both ME hardware for processing 4 MBs and on the average processing one MB requires 282 clock cycles. Both ME hardware can work at 113 MHz. Therefore, they are capable of processing 49 1920x1080 full HD frames per second.

The 1BT based ME hardware architectures proposed in this chapter are based on the ME hardware architecture proposed in [13]. However, the ME hardware proposed in [13] is performing 8 bits/pixel ME using SAD block matching criterion, it is implementing a Hexagon-Based ME algorithm, it is not performing ME for 4 MBs in parallel, and it is not performing VBS ME.

The comparison of the proposed FBS ME and VBS ME hardware with the Full Search ME hardware proposed in [4, 5, 14, 19] is shown in Table 3.1. The proposed 1BT based ME hardware architectures are faster and have less logic area and on-chip memory than the 8 bits/pixel VBS ME hardware architectures proposed in [4, 5].

We synthesized the 1BT based ME hardware architectures presented in [14, 19] using Mentor Graphics Precision RTL 2005b and mapped them to a Xilinx XC2VP30-7 FPGA using Xilinx ISE 8.2i. The 1BT ME hardware architecture proposed in [14] 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 [19] 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 architectures are larger than the area of the 1BT based ME hardware architectures proposed in [14, 19] because of performing ME for 4 MBs in parallel and data alignment. However, the proposed ME hardware architectures are much faster and have much less on-chip memory than these ME hardware architectures.

Table 3.1 Comparison of Motion Estimation Hardware Architectures

	Proposed (FBS)	**Proposed (VBS)**	**[14]**	**[19]**	**[4]**	**[5]**
Bit Depth	1	1	1	1	8	8
On-Chip SW Memory (bits)	4608	4608	24064	24064	26624	24192
Area	7280 LUTs 2745 DFFs	8702 LUTs 6401 DFFs	1589 LUTs 478 DFFs	1467 LUTs 499 DFFs	160K Gates	76400 LUTs 18000 DFFs
Maximum Frequency (MHz)	115	113	117	127	200	198
Technology	XC2VP30 FPGA	XC2VP30 FPGA	XC2VP30 FPGA	XC2VP30 FPGA	0.18μm ASIC	XC5VLX330 FPGA
Search Range	[-16, 16]	[-16, 16]	[-16, 15]	[-16, 15]	[-16, 16]	[±24, ±16]
Search locations / MB	1089	1089	1024	1024	1089	1584
Performance (1920x1080 fps)	50	49	13	15	21	31
Performance (1280x720 fps)	113	111	31	33	49	69
Supported MB Partitions	16x16	4x4, 4x8, 8x4, 8x8, 16x8, 8x16, 16x16	16x16	16x16	4x4, 4x8, 8x4, 8x8, 16x8, 8x16, 16x16	4x4, 4x8, 8x4, 8x8, 16x8, 8x16, 16x16

CHAPTER IV

HIGH PERFORMANCE HARDWARE ARCHITECTURES FOR ONE BIT TRANSFORM BASED MOTION ESTIMATION WITH SQUARE MACROBLOCK ORGANIZATION

4.1 Proposed Hardware Architecture for One Bit Transform based Fixed Block Size Single Reference Frame Motion Estimation

The proposed 1BT based FBS SRF ME 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. SWs of 4 16x16 MBs (MB0, MB1, MB2 and MB3) and their search locations for [0,0] MV are shown in Figure 4.1. SW_T size for 4 MBs is 64x64 pixels. There are large intersections between the SWs of these 4 MBs when they are organized in a square shape, e.g. 2/3 of the SWs of MB0 and MB1 are the same and 4/9 of the SWs of MB0 and MB3 are the same. Since 48x48 SW for a single MB requires loading 2304 bits to on-chip memory, processing 4 MBs requires loading 9216 bits to on-chip memory. However, performing ME for these 4 MBs in parallel with the proposed square shape organization allows significant data reuse and therefore only 4096 bits need to be loaded to on-chip memory.

The 1BT based FBS SRF ME hardware proposed in Chapter III also searches 4 MBs in parallel. However, in that ME hardware, the 4 MBs have a rectangular organization as shown in Figure 3.2. Therefore, the SWs of rightmost and leftmost MBs do not intersect, and this requires 96x48=4608 bits on-chip memory. The proposed ME hardware with square organization uses 11% less on-chip memory than the ME hardware proposed in Chapter III. In addition, the square organization simplifies the data alignment, and this reduces the logic area of the ME hardware.
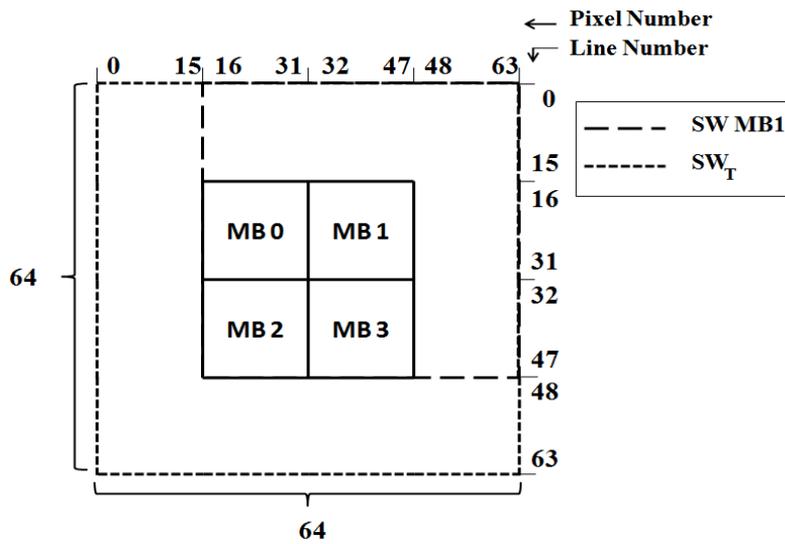
Figure 4.1 Search Windows of 4 MBs
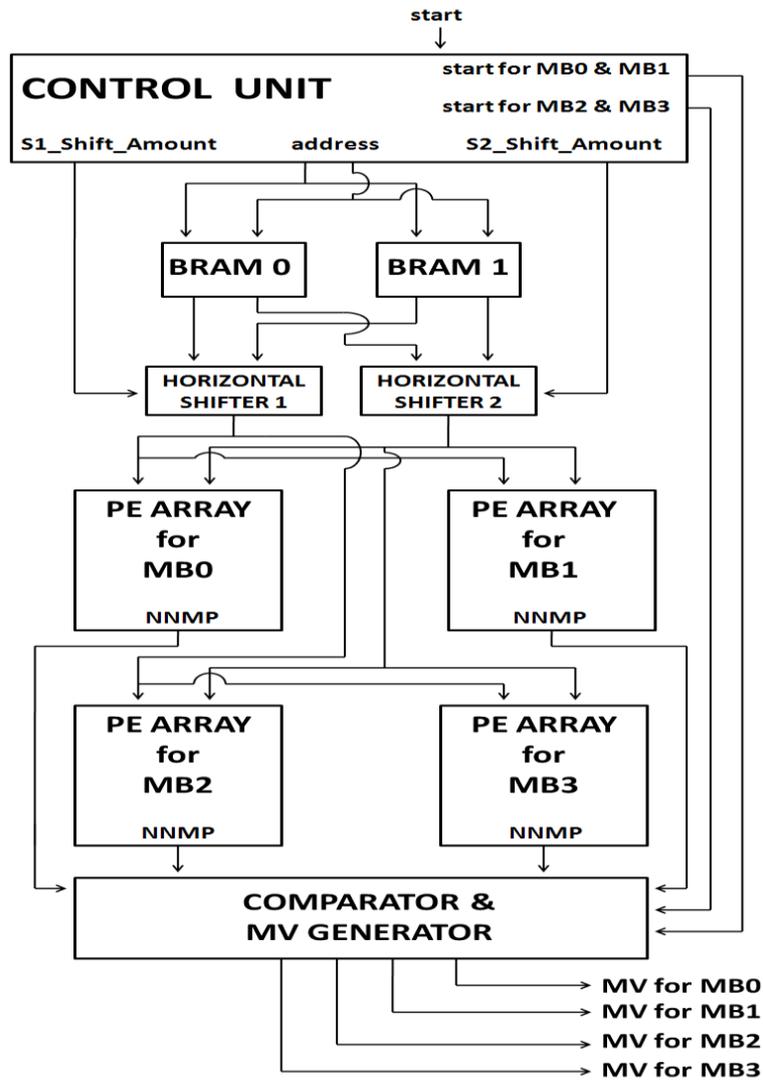
Figure 4.2 Top-Level Block Diagram of Proposed SRF ME Hardware

In the proposed 1BT based FBS SRF ME hardware, the search locations in a SW_T are searched column by column and the search locations in each column are searched from top to bottom. The first search locations searched by MB0 and MB1 includes the SW_T pixels 0 to 15 and 16 to 31 respectively in the lines 0 to 15. The first search locations searched by MB2 and MB3 includes the SW_T pixels 0 to 15 and 16 to 31 respectively in the lines 16 to 31.

The top-level block diagram of the proposed 1BT based FBS SRF ME hardware is shown in Figure 4.2. The hardware has 2 BRAMs, 2 Horizontal Shifters, 4 Processing Element (PE) arrays, Control Unit, and Comparator & MV Generator. Its latency is 6 clock cycles; 1 cycle for Control Unit, 1 cycle for synchronous read from memory, 1 cycle for Horizontal Shifter, 1 cycle for Non-Match Counter, 1 cycle for accumulation, and 1 cycle for Comparator & MV Generator.

There are 256 PEs in each PE array. 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. The architecture of the PE arrays for MB0 and MB2 is the same and it is shown in Figure 4.3. As it can be seen in the figure, the number of non-matching points accumulation is performed sequentially through the rows of the PE array. For any candidate search location, the latency between loading SW pixels to the PEs in the 1st row of PE Array and the PEs in the 16th row of PE array is 15 cycles. Because of the pipelining, NNMP values for the candidate search locations become available in every clock cycle after NNMP value for the [-16,-16] candidate search location is available. The architecture of the PE arrays for MB1 and MB3 is the same, and the only difference between this PE array architecture and the PE array architecture shown in Figure 4.3 is that the PEs in this architecture take the SW pixels from the 16 least significant bits of the 32-bit outputs of the horizontal shifters, instead of the 16 most significant bits.

The architecture of a PE is shown in Figure 4.4. Each PE performs XOR operation between a SW pixel and a current MB pixel. The result of an XOR operation indicates whether the SW pixel and the current MB pixel match or not. NNMP values for each row of the MB are computed by using Non-Match Counters as shown in Figure 4.5. The architecture of Non-Match Counters is presented in Figure 3.5 (b). It counts the ones in the outputs of 16 XOR gates by using 4 look up tables with 2^4 entries and adding the outputs of these look up tables. The results of these 16 Non-Match Counters are accumulated to compute the NNMP

value for a search location. Therefore, 64 Non-Match Counters are used in the proposed hardware for computing NNMPs of 4 MBs in parallel.
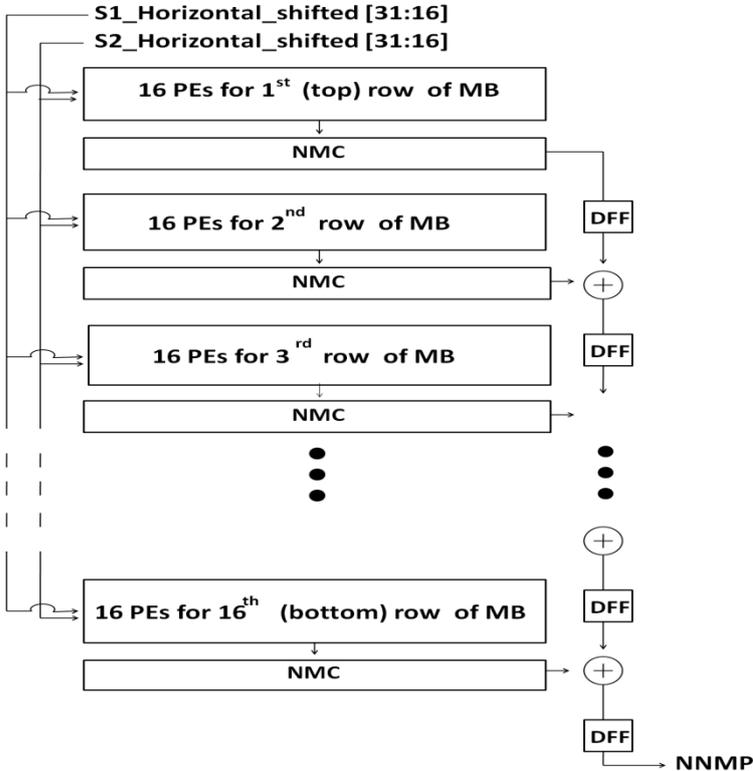
Figure 4.3 PE Array Architecture for MB0
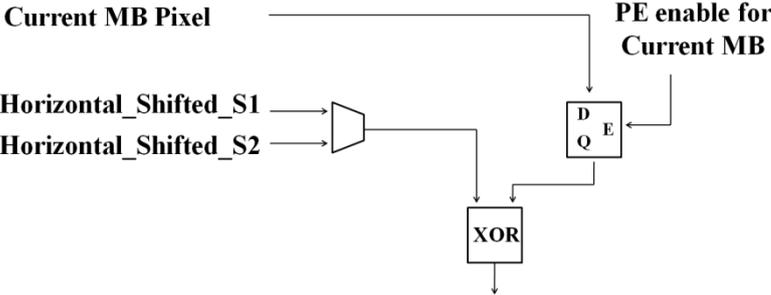
Figure 4.4 PE Architecture
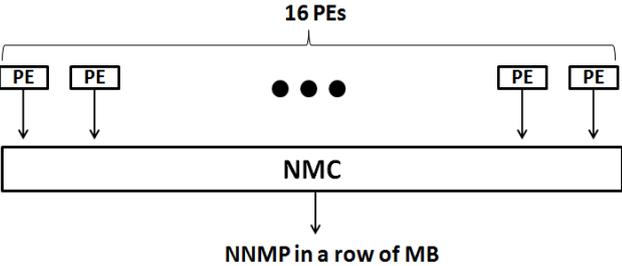
Figure 4.5 Connection between 16 PEs and NMC

The memory organization of the proposed 1BT based FBS SRF ME hardware is shown in Figure 4.6. 2 dual-port BRAMs in the FPGA are used for storing a 64x64 SW_T. 32-bits are stored in each address of a BRAM, and 32-bit output ports of BRAMs are named as S1 and S2. In the proposed SRF ME hardware, the candidate search locations pointed by same motion vectors are searched for MB0 and MB1, and the candidate search locations pointed by same motion vectors are searched for MB2 and MB3. Therefore, the same S1 and S2 address values are sent to the 2 BRAMs during the search process and 2 64-bit lines of a SW_T are read from BRAMs in each cycle.

The usage of the multiplexer shown in Figure 4.4 and the data flow of the proposed ME hardware are similar to the ones proposed in [19]. However, 1BT based ME hardware in [19] is not searching 4 MBs in parallel, and it is not using the data reuse proposed in this Chapter for reducing the memory usage.

Horizontal shifter is used to align the SW pixels coming from BRAMs. It shifts a 64-bit line of SW_T coming from BRAMs to the right, and rightmost 32-bits are used as input to the PE arrays. The PE Arrays for MB1 and MB3 take the least significant 16-bits of the outputs of Horizontal Shifters, and the PE Arrays for MB0 and MB2 take the most significant 16-bits of the outputs of Horizontal Shifters. The addresses for BRAMs and the horizontal shift amounts are shown in Table 4.1 for the first 100 clock cycles.
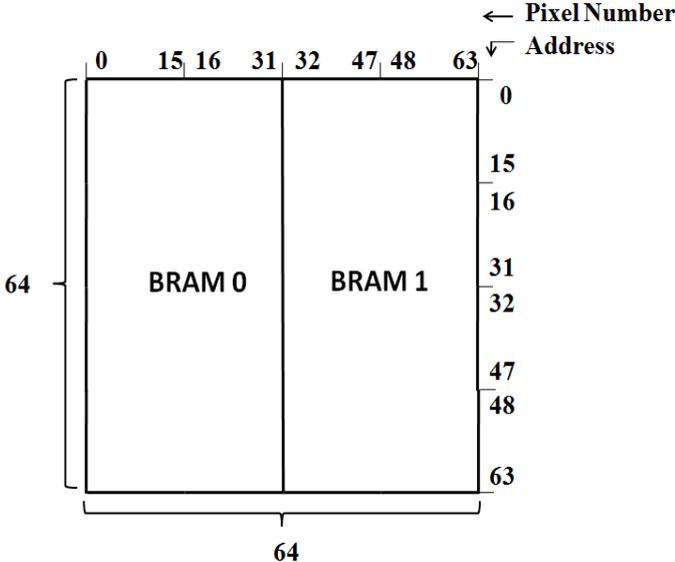
Figure 4.6 Memory Organization for Storing SW Pixels

Table 4.1 Control Signals for BRAMs and Horizontal Shifters

Clock Cycle	Addresses for BRAMs		Horizontal Shift Amount	
	S1	S2	S1	S2
0	0	–	32	–
1	1	–	32	–
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
32	32	–	32	–
33	33	0	32	31
34	34	1	32	31
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
63	63	30	32	31
64	–	31	–	31
65	–	32	–	31
66	0	33	30	31
67	1	34	30	31
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
96	30	63	30	31
97	31	–	30	–
98	32	–	30	–
99	33	0	30	29
100	34	1	30	29
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

4.2 Proposed 1BT Based Variable Block Size Single Reference Frame Motion Estimation Hardware

The top-level block diagram of the proposed VBS SRF ME hardware architecture is similar to the top-level block diagram of the proposed FBS SRF ME hardware architecture shown in Figure 4.2. The NMC and Adder Trees used in the PE arrays are different. MB0 PE array for VBS ME hardware is shown in Figure 4.7. As shown in Figure 4.3 and Figure 4.7, even though each PE array in FBS ME hardware computes the NNMP value for a MB, each PE array in VBS ME hardware computes the NNMP values for the 41 partitions of a MB. The 41 partitions of a MB are shown in Figure 3.10.

In VBS ME hardware, an NMC in a PE array computes the NNMP value for 4 current MB and 4 SW pixels using a look up table with 2^4 entries. In the Adder Tree, the outputs of the NMCs are added to compute the NNMP values for the 16 4x4 blocks. For example, NMC (0, 0), NMC (1, 0), NMC (2, 0) and NMC (3, 0) are added to compute the NNMP value for 4x4 block 1 as shown in Figure 3.10.
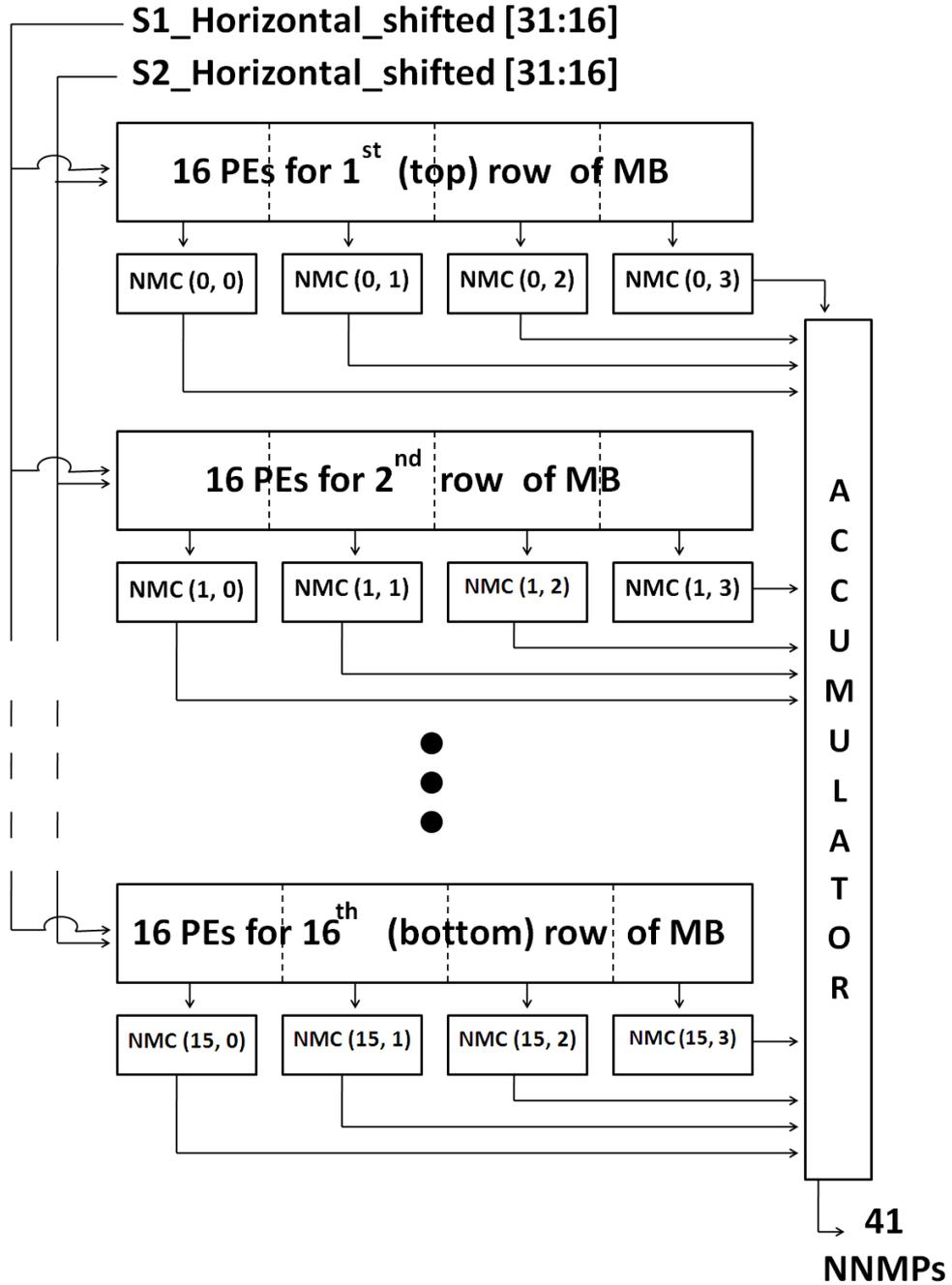
Figure 4.7 MB0 PE Array Architecture for VBS ME

SW_T pixels are read from the BRAMs row by row. For a search location, the NNMP values for the 4x4 blocks are calculated in 16 clock cycles. The NNMP values for the blocks 1, 2, 3, 4 are calculated in the first 4 clock cycles, the NNMP values for the blocks 5, 6, 7, 8 are calculated in the next 4 clock cycles, the NNMP values for the blocks 9, 10, 11, 12 are calculated in the next 4 clock cycles, and the NNMP values for the blocks 13, 14, 15, 16 are calculated in the last 4 clock cycles.

The NNMP values for the 8x4 and 4x8 blocks are calculated by adding the NNMP values of the corresponding 4x4 blocks as they become available. The NNMP values for the 8x8 blocks are computed by adding the NNMP values of the corresponding 4x8 blocks as they become available. The NNMP values for the 8x16 and 16x8 blocks are computed by adding the NNMP values of the corresponding 8x8 blocks as they become available. The NNMP value for the 16x16 block is computed by adding the NNMP values of the 8x16 blocks as they become available.

The 41 NNMP values of a 16x16 MB are calculated in 16 clock cycles. The 41 NNMP values of each MB computed by a PE array are sent to Comparator & MV Generator, and the Comparator & MV Generator determines the minimum NNMP values and the corresponding MVs for each MB partition.

4.3 One Bit Transform Based Multiple Reference Frame Motion Estimation Algorithm

In 1BT based MRF ME, NNMP values of the candidate search locations in the SWs of all RFs are compared, and the search location that gives the minimum NNMP is selected as the best match.

The number of operations performed per pixel (pp) by the 1BT based ME methods implementing FS algorithm with a block size of 16x16 pixels and a search range of [-16, 16] are shown in Table 4.2. The numbers of previous and future RFs are shown as MRF-1BT$_{-previous+future}$. The kernel proposed in [15] is used for the 1BT based MRF ME, therefore MRF-1BT$_{-1+0}$ is same as MF1BT. Bool. is an XOR operation, Comp. is an 8 bit comparison, and Inc. is an increment operation used for counting non-matching pixels from 0 to 256. For transform, MRF-1BT performs same number of operations as MF1BT, because each transformed one bit depth image is used multiple times for search. However, MRF-1BT requires larger off-chip memory.

The PSNR results of FS ME with 8-bit depth matching, MF1BT and 1BT based MRF ME for a block size of 16x16 pixels and a search range of [-16, 16] are compared for various video sequences in Table 4.3. The PSNR values in dB are calculated between the original

frames and frames reconstructed from the RFs using the MVs calculated by these ME methods. As it can be seen from Table 4.3, the PSNR results of MRF-1BT$_{-1+1}$, MRF-1BT$_{-2+0}$ and MRF-1BT$_{-0+2}$ are considerably better than the PSNR result of MRF-1BT$_{-1+0}$. The PSNR result of MRF-1BT$_{-2+2}$ is only slightly better than the PSNR result of MRF-1BT$_{-1+1}$.

Table 4.2 Number of Operations for a Search Range of [-16, +16]

ME Method	TRANSFORM			MATCHING			MEMORY	
	13bit + 8bit Addition (pp)	Shift (pp)	8 bit Comp. (pp)	1bit Bool. (pp)	8bit + 1bit Inc. (pp)	8 bit Comp. (pp)	Off-Chip (bits) (pp)	On-Chip (bits) (pp)
MF1BT[15] MRF-1BT$_{-1+0}$	16	1	1	1089	1089	4.25	9	1
MRF-1BT$_{-1+1}$	16	1	1	2178	2178	8.51	10	2
MRF-1BT$_{-2+0}$	16	1	1	2178	2178	8.51	10	2
MRF-1BT$_{-2+2}$	16	1	1	4356	4356	17.01	12	4

Table 4.3 Average PSNR for Several Video Sequences

ME Method	Video Sequence (Frame Size) (Sequence Length)						Average PSNR Improvement (%) over MF1BT [15]
	Football (352x240) (150 frames)	Foreman (352x288) (150 frames)	Tennis (352x240) (150 frames)	Susie (352x240) (150 frames)	Mobile (352x240) (150 frames)	Coastguard (352x288) (150 frames)	
MF1BT[15] (MRF-1BT$_{-1+0}$)	22.25	31.81	30.21	33.22	22.74	25.92	0.00
8-bit depth FS	23.32	33.29	31.12	34.26	23.10	26.56	3.27
MRF-1BT$_{-2+0}$	22.65	32.42	30.43	33.83	23.44	25.98	1.60
MRF-1BT$_{-0+2}$	22.61	32.36	30.38	33.68	23.41	25.96	1.40
MRF-1BT$_{-1+1}$	23.57	33.35	31.34	34.27	23.71	26.64	4.11
MRF-1BT$_{-2+2}$	23.61	33.94	31.40	34.81	24.45	26.74	5.37
MRF-1BT$_{-3+3}$	23.63	34.02	31.39	34.82	24.97	26.79	5.84
MRF-1BT$_{-4+4}$	23.56	33.90	31.31	34.66	25.10	26.81	5.71
MRF-1BT$_{-5+5}$	23.54	33.85	31.34	34.66	25.05	26.80	5.64

In addition, MRF-1BT$_{-1+1}$ and MRF-1BT$_{-2+2}$ have better PSNR results than 8-bit depth FS ME, although they have considerably less computational complexity. 8-bit depth FS ME requires 1089 8-bit absolute difference operations, 1089 16-bit accumulation, 4 comparisons, 8-bit on-chip memory and 8-bit off-chip memory per pixel.

The results also show that, because of low bit depth representation of pixels in 1BT, if a large number of previous and future frames (e.g. MRF-1BT$_{-5+5}$) are used in search process, wrong motion vectors can be found and this reduces ME performance. Based on the results in

Table 4.3, it can be concluded that using up to 2 previous and 2 future frames provides a good trade-off between ME performance and computational complexity.

4.4 Proposed Reconfigurable Hardware Architecture for 1BT Based Multiple Reference Frame Motion Estimation

In this thesis, we also propose a 1BT based reconfigurable FBS MRF ME hardware based on the proposed 1BT based FBS SRF ME hardware. The MRF ME hardware has more logic area and uses more memory than the FBS SRF ME hardware. But, it has the same speed since it processes 4 MBs and 4 RFs (RF1, RF2, RF3, RF4) in parallel. The top-level block diagram of the proposed reconfigurable 1BT based MRF ME hardware is shown in Figure 4.8. It has 8 BRAMs, 8 Horizontal Shifters, 4 PE arrays, Control Unit, and Comparator & MV Generator.

The datapaths of MRF ME and FBS SRF ME hardware architectures are similar. The only difference is that the datapath of MRF ME hardware processes multiple RFs in parallel. Therefore, MRF ME hardware architecture uses 8 BRAMs and 8 Horizontal Shifters instead of 2 BRAMs and 2 Horizontal Shifters. However, because of the parallel processing, the memory organization shown in Figure 4.6 and control signals shown in Table 4.1 are the same for the SRF and MRF ME hardware architectures. In the proposed MRF ME hardware, the same candidate search locations are searched in all RFs. Therefore, the same address values and shift amount signals are sent to all BRAMs and Horizontal Shifters during the search process.

Same as the FBS SRF ME hardware, in MRF ME hardware there are 256 PEs in each PE array. However MRF ME hardware requires more complex accumulation hardware and more non-match counters. The architecture of the PE arrays for MB0 and MB2 is the same and it is shown in Figure 4.9. The architecture of the PE arrays for MB1 and MB3 is the same, and the only difference between this PE array architecture and the PE array architecture shown in Figure 4.9 is that the PEs in this architecture take the SW pixels from the 16 least significant bits of the 32-bit outputs of the horizontal shifters, instead of the 16 most significant bits.
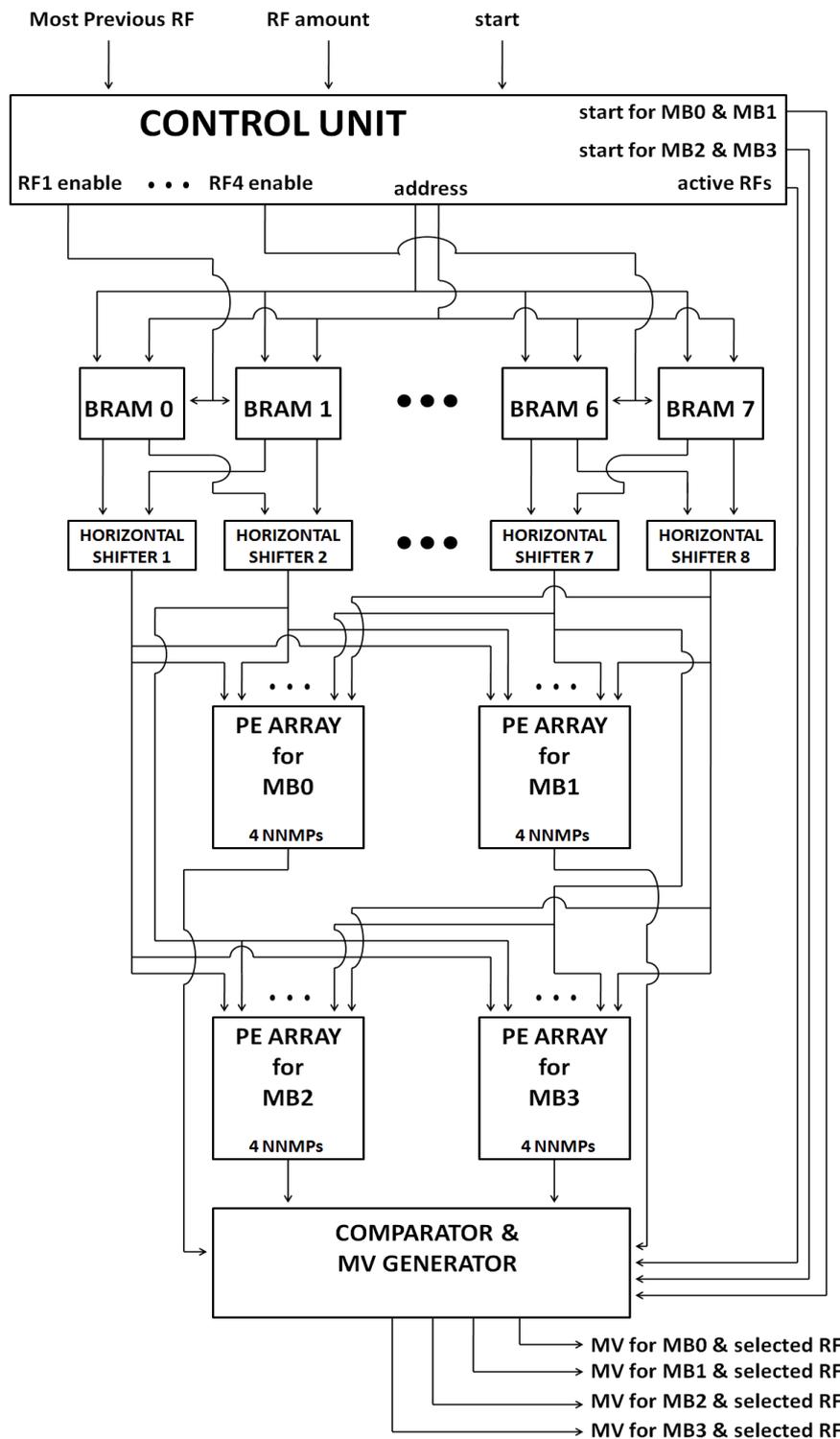
Figure 4.8 Top-Level Block Diagram of Proposed MRF ME Hardware
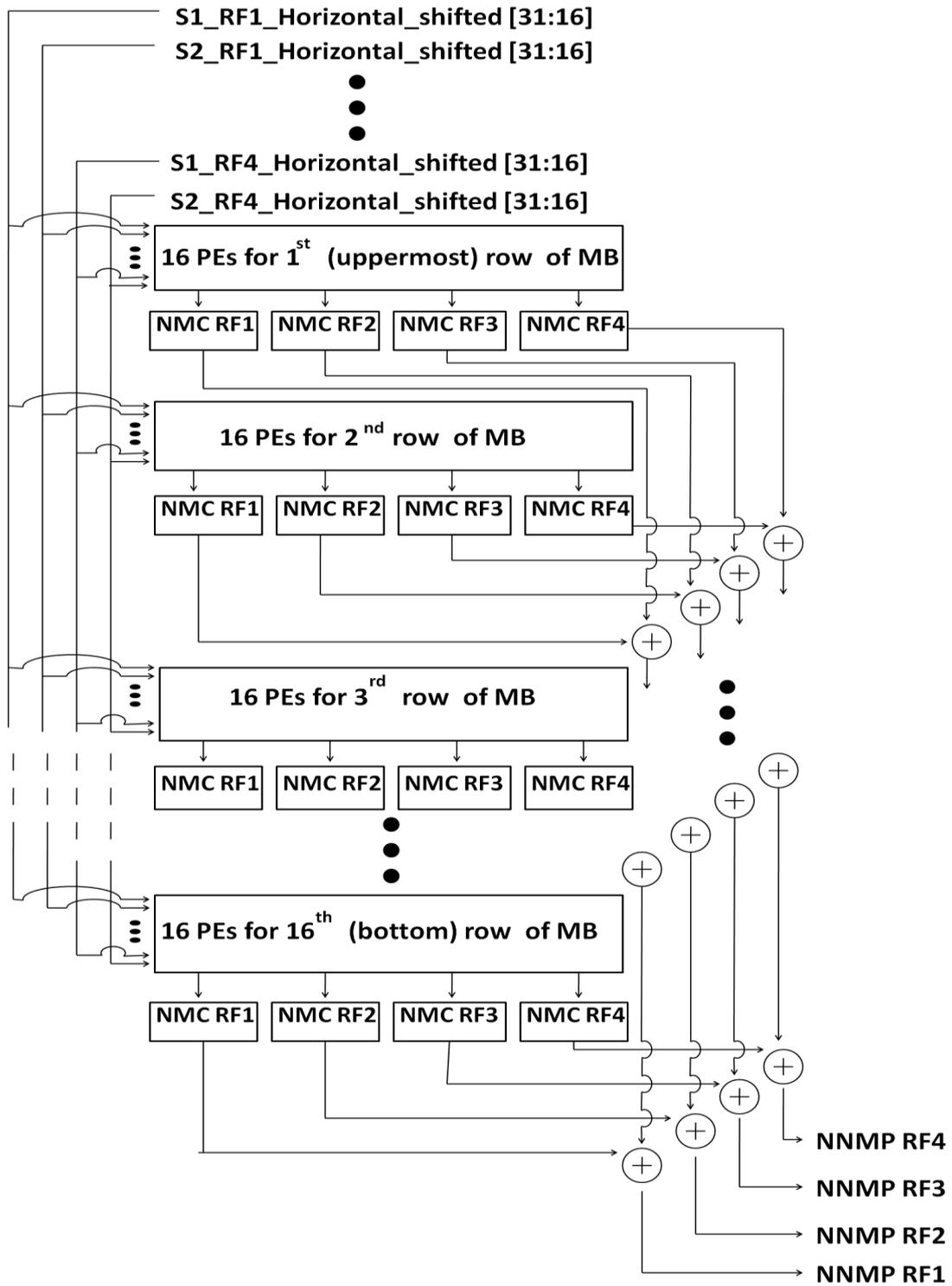
Figure 4.9 MRF ME PE Array Architecture for MB0

In MRF ME hardware, each PE array computes 1 NNMP value for each RF. Therefore, the NMCs and the accumulators in MRF ME hardware have 4 times more logic area than the ones in FBS SRF ME hardware. The architecture of a PE in MRF ME hardware

is shown in Figure 4.10. Each PE performs 4 XOR operations between SW pixels coming from 4 RFs and a current MB pixel.

NNMP values for each row of a MB are computed by using NMCs as shown in Figure 4.11. In the proposed FBS SRF ME hardware, 64 NMCs are used for computing NNMPs of 4 MBs in parallel. Therefore, in MRF ME hardware, 256 NMCs are used for searching 4 RFs in parallel.

Since MRF ME hardware is searching 4 RFs in parallel, the Comparator & MV Generator in MRF ME hardware is more complex than the one in SRF ME hardware. In order to make the clock frequency of MRF ME hardware same as the clock frequency of SRF ME hardware, pipeline latency of Comparator & MV Generator is increased from 1 to 3.
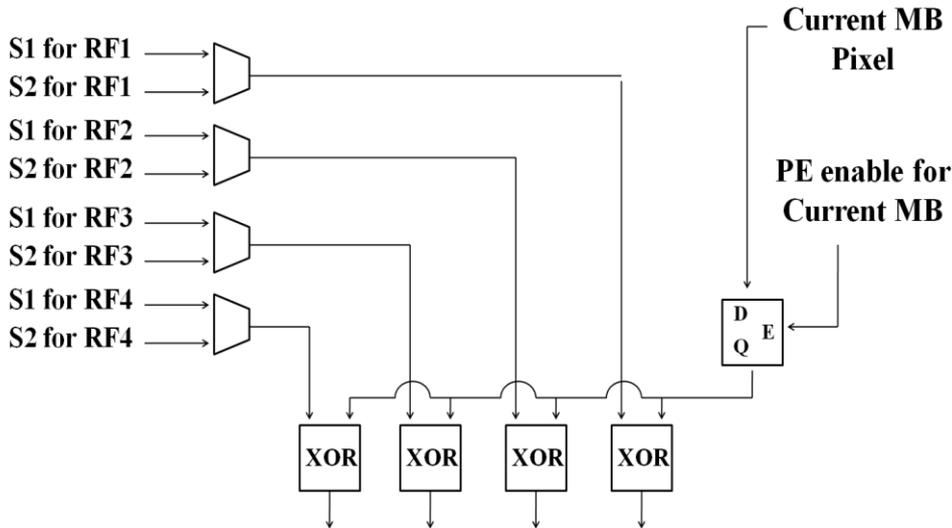
Figure 4.10 PE Architecture of MRF ME Hardware
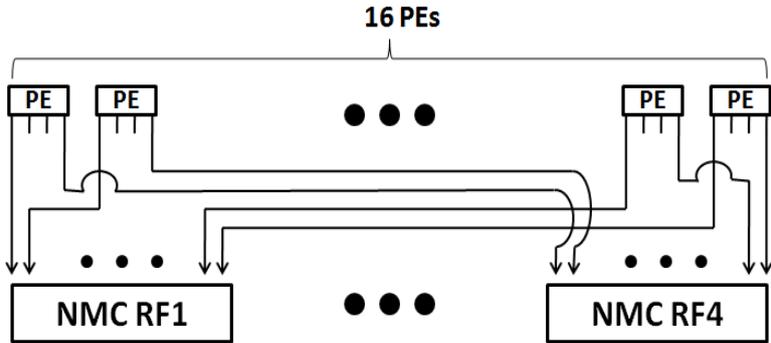
Figure 4.11 Connection between 16 PEs and 4 NMCs

In proposed MRF ME hardware architecture, the candidate RFs for every 4 MBs can be reconfigured by the main controller depending on application requirements. Control Unit takes the candidate RFs with two inputs, *most previous RF* and *RF amount*. For example, MRF-1BT$_{-2+1}$ configuration can be used for a frame rate up conversion application, and it can be changed to MRF-1BT$_{-2+2}$ by setting *most previous RF* to *-2* and *RF amount* to *4* during the search process of the next 4 MBs in order to obtain better performance. As another example, MRF-1BT$_{-3+0}$ configuration can be used for a video compression application, and it can be changed to MRF-1BT$_{-1+0}$ during the search process of the next 4 MBs in order to reduce the computational complexity.

Control Unit determines the codes of the RFs (RF1, RF2, RF3, RF4) and sends them to Comparator & MV Generator. For example, for the MRF-1BT$_{-3+1}$ configuration, the codes of RF1, RF2, RF3, and RF4 are *-3, -2, -1* and *1* respectively. Comparator & MV Generator compares the NNMP values of the candidate search locations in the candidate RFs, and stores the minimum NNMP value, candidate MV and the code of the corresponding RF.

If less than 4 RFs are used to reduce computational complexity, Control Unit disables the BRAMs of the unused RFs by changing *RF enable* signals shown in Figure 4.8 and sends *0* as the codes of the unused RFs. Then, during the search process, Comparator & MV Generator uses maximum NNMP value for the *0* coded RFs in order to avoid selecting their MVs. Disabling BRAMs and using maximum NNMP value for the comparison stops the switching activity in Horizontal Shifters, PE Arrays and Comparator & MV Generator.

4.5 Implementation Results

The proposed 1BT based FBS SRF ME, VBS SRF ME and MRF ME hardware architectures are implemented in Verilog HDL. The Verilog RTL codes are mapped to a XC2VP30-7 FPGA using ISE 8.2i. The hardware implementations are verified with post place & route simulations using Modelsim 6.1c.

The FBS SRF ME hardware consumes 2642 slices (3914 LUTs), which is 19% of all the slices in XC2VP30-7 FPGA. In addition, 4096 bits on-chip memory is used for storing SWs of 4 MBs, and these 4096 bits are stored in 64 addresses of 2 BRAMs. The VBS SRF ME hardware consumes 4834 slices (4957 LUTs), which is 35% of all the slices in XC2VP30-7 FPGA. In addition, it uses 4096 bits on-chip memory, and these 4096 bits are stored in 64 addresses of 2 BRAMs. The MRF ME hardware consumes 9012 slices (15545 LUTs) which is 66% of all the slices in XC2VP30-7 FPGA. In addition, 16384 bits on-chip memory is used for storing SWs of 4 MBs of 4RFs, and these 16384 bits are stored in 64 addresses of 8 BRAMs.

1127 clock cycles are required by the proposed FBS SRF ME and VBS SRF ME hardware for processing 4 MBs. Therefore, on the average, processing one MB requires 282 clock cycles. MRF ME hardware has two additional pipeline stages for Comparator & MV Generator. Therefore, it requires 1129 clock cycles for searching 4 MBs in 4 RFs. The proposed ME hardware implementations can work at 191 MHz. Therefore, they are capable of processing 83 1920x1080 full HD frames per second.

The comparison of the proposed 1BT based FBS SRF ME, VBS SRF ME and MRF ME hardware with the 1BT based SRF ME hardware proposed in [14, 19, 23] and with the 8-bit depth ME hardware proposed in [5, 13] is shown in Table 4.4. We implemented and mapped the 1BT based ME hardware architectures presented in [14, 19, 23] to a XC2VP30-7 FPGA using Precision RTL 2005b and Xilinx ISE 8.2i. The proposed 1BT based ME hardware architectures are faster and have less logic area and on-chip memory than the 8 bits/pixel ME hardware architectures proposed in [5, 13].

The proposed 1BT based FBS SRF ME hardware has 11% less on-chip memory, 46% less LUTs and 8% less DFFs than the best 1BT based FBS SRF ME hardware presented in the literature [23]. It is also 68% faster than that FBS SRF ME hardware even though they are both processing 4 MBs in parallel. The reason for the reduction in on-chip memory usage is that the square (2x2) organization of the 4 MBs increases the intersections of their SWs in comparison to the rectangular (1x4) organization proposed in Chapter III.

Table 4.4 Comparison of Motion Estimation Hardware Architectures

	Proposed (SRF)	Proposed (MRF)	Proposed (VBS)	[14]	[19]	[23] (FBS)	[23] (VBS)	[13]	[5]
Bit Depth	1	1	1	1	1	1	1	8	8
On-Chip SW Memory (bits)	4096	16384	4096	24064	24064	4608	4608	57344	24192
AREA LUTs	3914	15545	4957	1589	1467	7280	8702	9128 Slices	76400
AREA DFFs	2517	6668	6049	478	499	2745	6401		18000
Maximum Frequency (MHz)	192	191	192	117	127	115	113	130	198
Technology	XC2VP30 FPGA	XC2VP30 FPGA	XC2VP30 FPGA	XC2VP30 FPGA	XC2VP30 FPGA	XC2VP30 FPGA	XC2VP30 FPGA	XC3S1500-5 FPGA	XC5VLX330 FPGA
Search Range	[-16, 16]	[-16, 16]	[-16, 16]	[-16, 15]	[-16, 15]	[-16, 16]	[-16, 16]	[±48, ±24]	[±24, ±16]
Search locations / MB	1089	4356	1089	1024	1024	1089	1089	242	1584
Performance (1920x1080 fps)	84	83	84	13	15	50	49	34	31
Performance (1280x720 fps)	189	188	189	31	33	113	111	77	69
Number of Reference Frames	1	4	1	1	1	1	1	1	1
Supported MB partitions	16x16	16x16	4x4, 4x8, 8x4, 8x8, 16x8, 8x16, 16x16	16x16	16x16	16x16	4x4, 4x8, 8x4, 8x8, 16x8, 8x16, 16x16	16x16	4x4, 4x8, 8x4, 8x8, 16x8, 8x16, 16x16

In addition, data alignment scheme used in the hardware presented in Chapter III requires Vertical Rotator (2048 LUTs) and One Bit Selector (256 LUTs). These are replaced with 2 Horizontal Shifters (2x128 LUTs) in the proposed FBS SRF ME hardware. The replacement of Vertical Rotator and One Bit Selector with Horizontal Shifters also removes the longest path of the hardware proposed in Chapter III, and therefore results in higher speed. In addition, the accumulation process in the proposed FBS SRF ME hardware requires less area than the Adder Tree used in the hardware proposed in Chapter III.

The logic area of the proposed 1BT based FBS SRF ME hardware is larger than the logic area of the 1BT based FBS SRF ME hardware proposed in [14, 19] because of performing ME for 4 MBs in parallel and data alignment. However, the proposed FBS SRF

ME hardware architecture is much faster and uses much less on-chip memory than those FBS SRF ME hardware architectures.

1BT based FBS SRF ME hardware architectures proposed in [14, 19] implement FS algorithm for a [-16, 15] search range and a 16x16 MB size. This requires storing 47x47 pixel = 2209 bits SW in on-chip memory. However, these ME hardware 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. 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. If 64 bits can be loaded into on-chip memory from off-chip memory in each clock cycle, 24064 bits on-chip memory for the SW of one MB can be loaded in 376 clock cycles.

However, the proposed FBS SRF ME hardware requires much less on-chip memory for searching 4 MBs. If 64 bits can be loaded into on-chip memory from off-chip memory in each clock cycle, 4096 bits on-chip memory can be loaded in 64 clock cycles. Therefore, the proposed FBS SRF ME hardware architecture needs on the average 16 clock cycles for loading the SW pixels of a MB instead of 376 clock cycles.

The proposed 1BT based VBS SRF ME hardware has 11% less on-chip memory, 43% less LUTs and 5% less DFFs than the 1BT based VBS SRF ME hardware presented in Chapter III. It is also 71% faster than that VBS SRF ME hardware even though they are both processing 4 MBs in parallel.

The area of the proposed 1BT based MRF ME hardware is larger than the area of the 1BT based SRF ME hardware proposed in [14, 19, 23], because of performing ME for 4 MBs in 4 RFs in parallel. However, it is faster than those ME hardware. In addition, although it is processing 4 MBs in 4 RFs in parallel, it uses less on-chip memory than the SRF ME hardware proposed in [14, 19].

CHAPTER V

HIGH PERFORMANCE HARDWARE ARCHITECTURE FOR EARLY TERMINATED CONSTRAINT ONE BIT TRANSFORM MOTION ESTIMATION

5.1 Proposed Hardware Architecture for Constraint One Bit Transform Motion Estimation Algorithm

The proposed C-1BT ME hardware is based on the FBS SRF ME hardware proposed in Chapter IV. C-1BT ME hardware is more complex and has higher on-chip memory requirement than the FBS SRF ME hardware. Same as the FBS SRF ME hardware, C-1BT ME hardware finds MVs of 4 16x16 MBs in parallel using full search ME algorithm. Therefore, their performances are the same. However, C-1BT ME hardware is based on minimum CNNMP criterion instead of NNMP criterion.

The block diagram of the proposed hardware for C-1BT ME is shown in Figure 5.1. The datapath of the proposed C-1BT ME hardware is very similar to the FBS SRF ME hardware proposed in Chapter IV. As shown in Figure 5.1, it searches 4 MBs in parallel. The FBS SRF ME hardware proposed in Chapter IV searches current MB only in a 1 bit depth SW. But, the C-1BT ME hardware also requires searching in CM SW. Therefore, C-1BT ME hardware requires more on-chip memory. As shown in Figure 5.1, the number of BRAMs and horizontal shifters is increased from 2 to 4. In the proposed C-1BT ME hardware, the same search locations in the CM and 1 bit depth image are searched in parallel. Therefore, control signals used for BRAMs and Horizontal Shifters are the same as the control signals shown in Table 4.1.
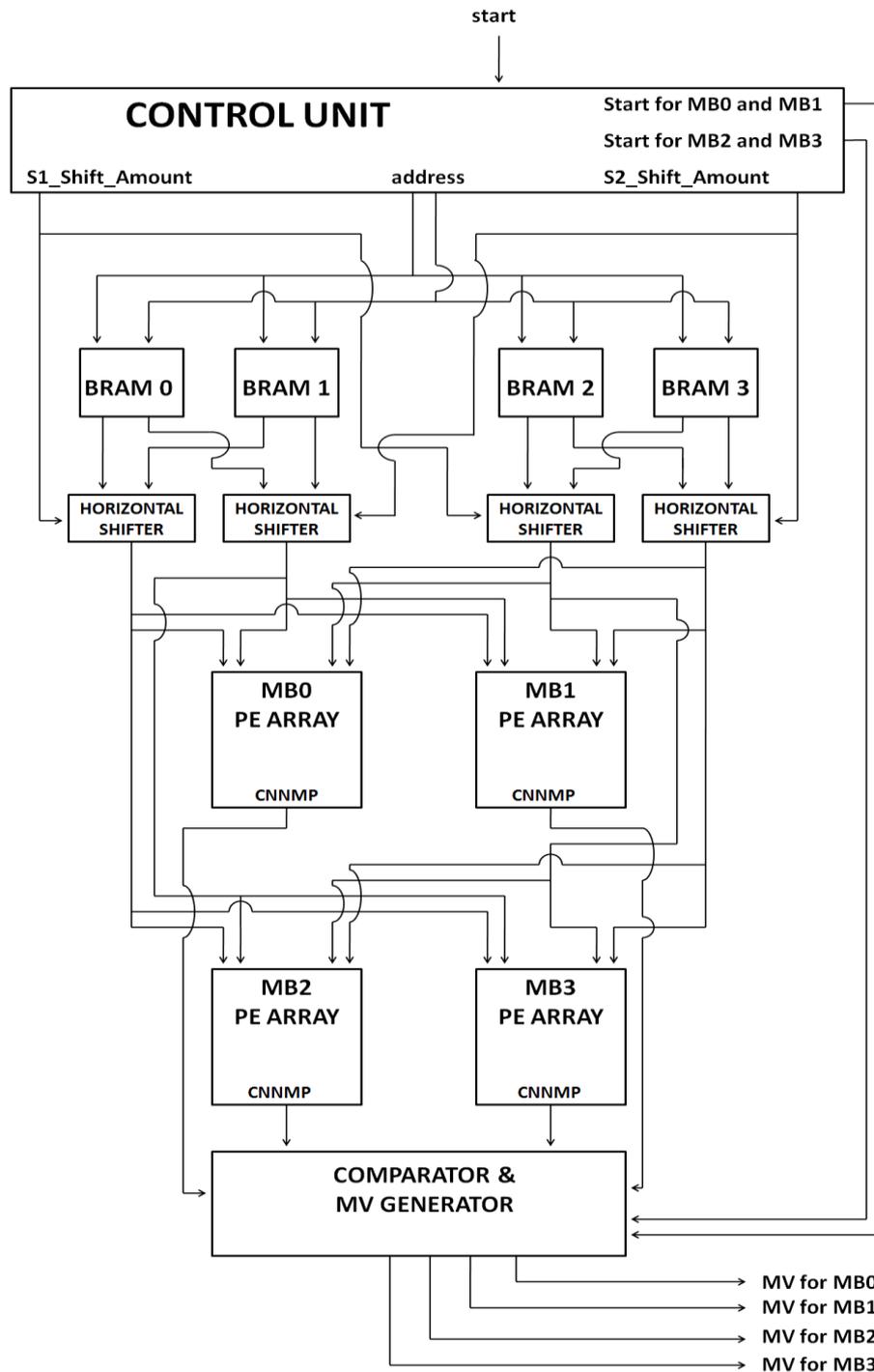
**Figure 5.1** Top-Level Block Diagram of the Proposed C-1BT ME Hardware

Same as the FBS SRF ME hardware proposed in Chapter IV, C-1BT ME hardware uses 256 PEs. But, the PE architectures are different. PE architecture proposed in Chapter IV performs an XOR operation between two 1 bit depth pixels. However, PEs in the C-1BT ME hardware perform logic operations between both CM values and 1 bit depth pixels. In C-1BT

ME hardware, the results of (XOR-OR-AND) operations performed by all 256 PEs in a PE array for a search location in the SW should be added to compute the CNNMP value for that search location. The architecture of the PE arrays for MB0 and MB2 is the same and it is shown in Figure 5.2. As it can be seen in the figure, the number of non-matching points accumulation is performed sequentially through the rows of the PE array. In C-1BT ME hardware, both the CM values and 1 bit depth pixels are used during the search process. Therefore, CM values are sent to the PE arrays together with the 1 bit depth pixels. This is the main difference between the PE arrays shown in Figure 4.3 and the PE arrays in C-1BT ME hardware.
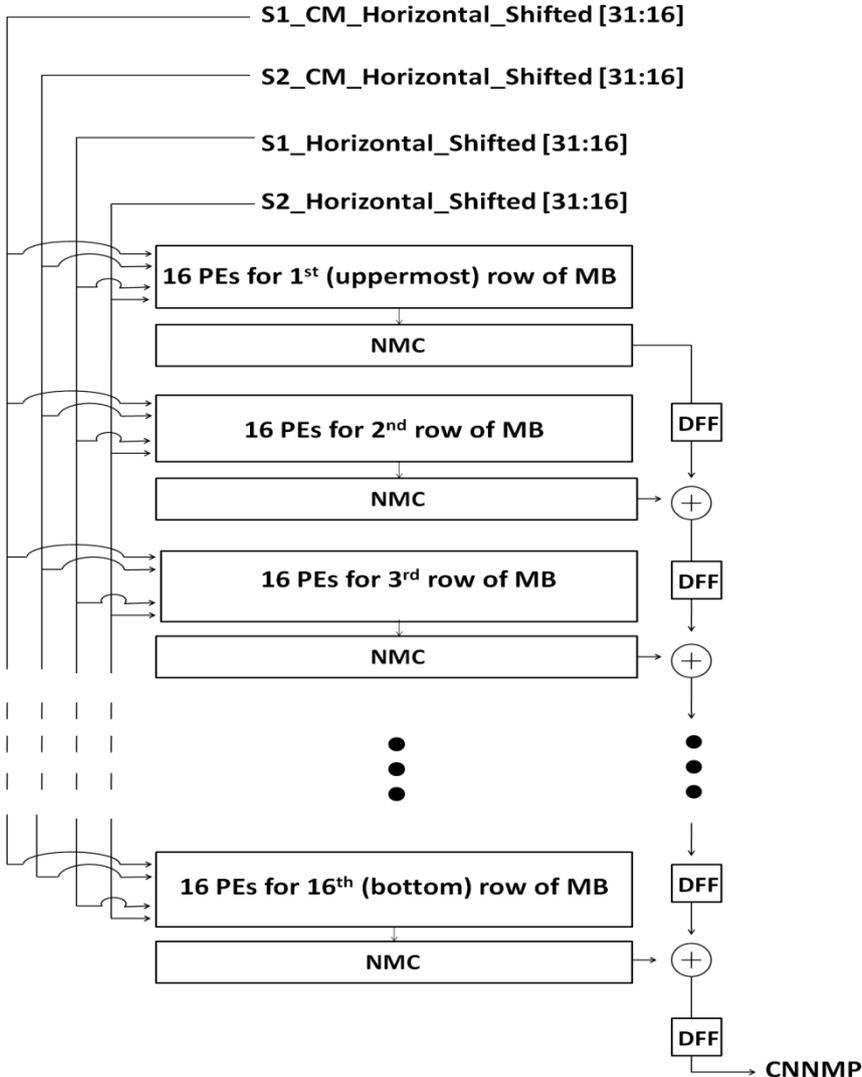

Figure 5.2 MB0 PE Array for C-1BT ME Hardware

The architecture of a PE used in C-1BT ME hardware is shown in Figure 5.3. Each PE performs XOR operation between a SW pixel from the 1 bit depth reference image and a current MB pixel from the 1 bit depth current image, OR operation between a corresponding SW value from the reference CM and a corresponding current MB value from the current CM, and then AND operation between the results of XOR and OR operations. The result of the AND operation indicates whether the SW pixel and the current MB pixel match or not. CNNMP values for each row of the MB are computed by using Non-Match Counters as shown in Figure 4.5. The architecture of an NMC is presented in Figure 3.5 (b). It counts the ones in the outputs of 16 AND gates by using 4 look up tables with 2^4 entries and adding the outputs of these look up tables. The results of these 16 NMCs are accumulated to compute the CNNMP value for a search location. Therefore, 64 NMCs are used in the proposed C-1BT ME hardware for computing CNNMP values of 4 MBs in parallel.
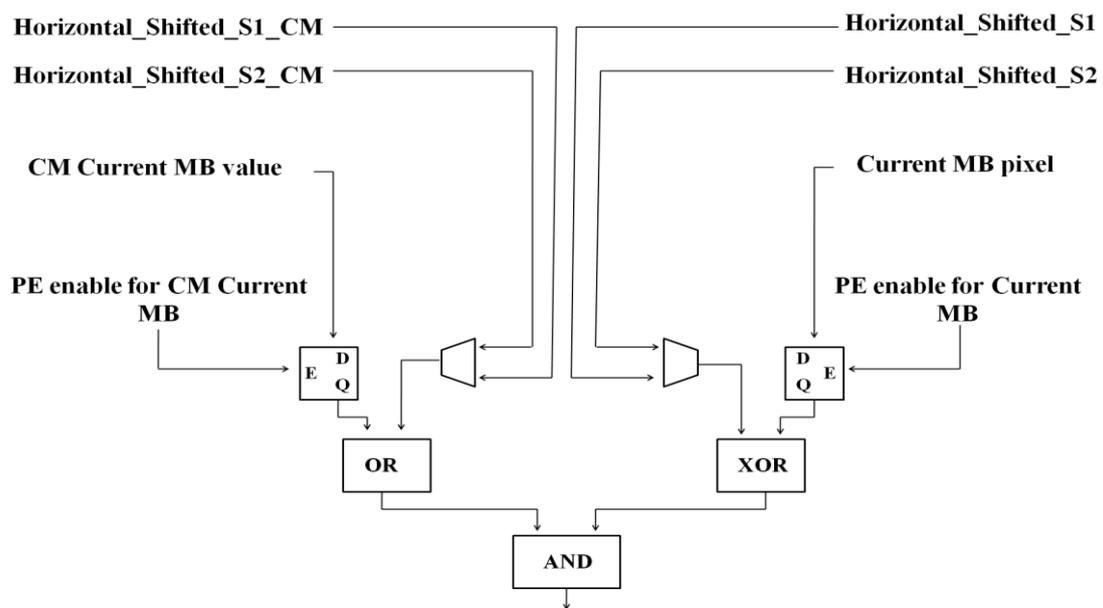
Figure 5.3 PE Architecture

Before ME starts for a MB, SW_T pixels from the 1 bit depth reference image are loaded to the 2 dual-port BRAMs (BRAM0 and BRAM1) in the FPGA, and the corresponding SW_T values from the reference CM are loaded to the 2 dual-port BRAMs (BRAM2 and BRAM3) in the FPGA. The memory organization of the 64x64 SW_T pixels from the 1 bit depth reference image is shown in Figure 4.6. The memory organization of the 64x64 SW_T values from the CM is similar to this memory organization. Left 32 bits of the CM values are stored in the BRAM2 and right 32 bits are stored in BRAM3.

5.2 Proposed Hardware Architecture for Early Terminated Constraint One Bit Transform Motion Estimation Algorithm

We modified the proposed C-1BT ME hardware for implementing early terminated C-1BT ME algorithm. The modifications in C-1BT ME hardware can be seen in Figures 5.4, 5.5, 5.6 and 5.7. As it shown in Figure 5.4, Early Termination Decision hardware is added to the C-1BT ME hardware. The modifications in PE Arrays are shown in Figure 5.5. The Early Termination Decision hardware architecture is shown in Figure 5.6. The modifications in Comparator & MV Generator are shown in Figure 5.7.
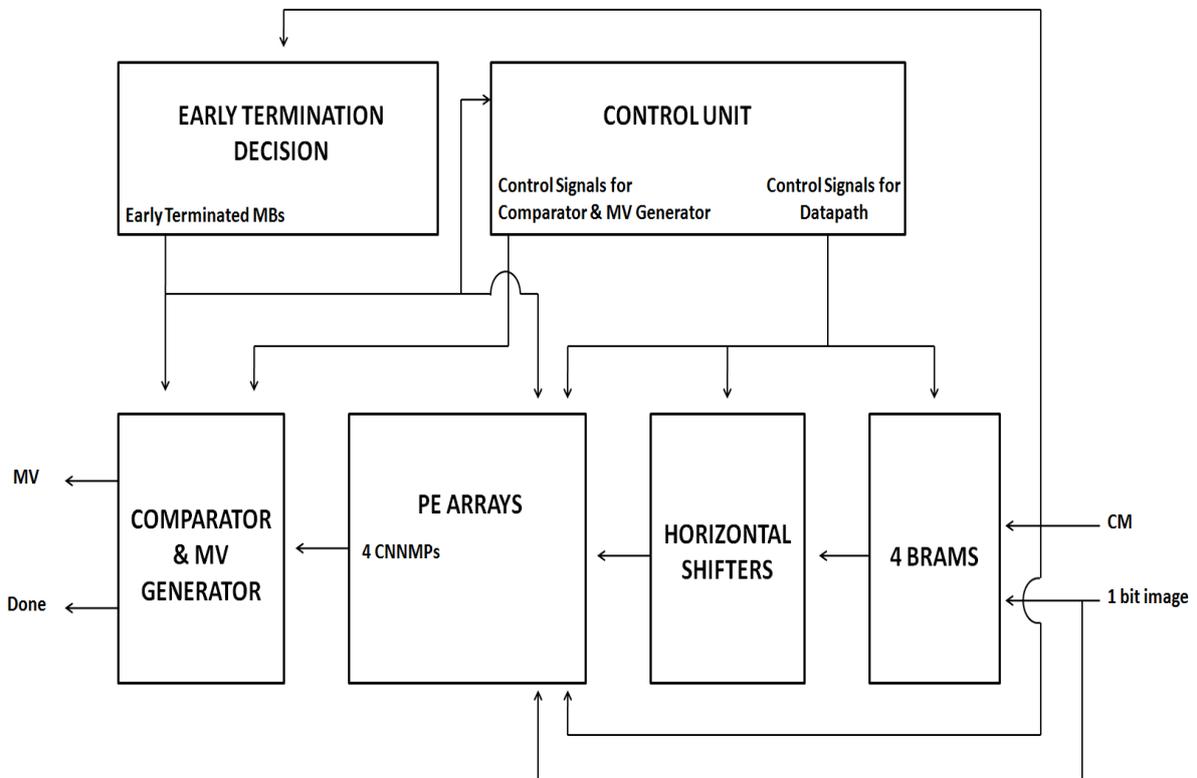
Figure 5.4 Top-level Block Diagram of the Proposed Early Terminated C-1BT ME Hardware

Early Termination Decision hardware starts working during data loading, before search process starts. While the current MB CM values of the 4MBs are loaded from the off-chip memory to the PE arrays, Early Termination Decision hardware also takes these values and starts determining early terminated MBs. It decides the early terminated MBs in 4 MBs before loading SW_T pixels from the off-chip memory to the BRAMs finishes. After it decides the early terminated MBs, it sends this information to Comparator & MV Generator, Control

Unit and PE array, and it avoids switching activities for the early terminated MBs until the data loading for the next 4 MBs starts.

It is also possible to determine the early terminated MBs in an entire image and then start search process for all the MBs in the image. However, this requires additional memory for storing the early termination information for each MB, i.e. whether these MBs are early terminated or not. In addition, this requires loading the CM values twice from the off-chip memory because CM values are used in both Early Termination Decision and search process, and this additional loading will cause additional power consumption.

The proposed Early Terminated C-1BT ME hardware searches 4 MBs in parallel. Therefore, it increases the performance of the C-1BT ME hardware when all of these 4 MBs are early terminated. If Early Termination Decision hardware determines that all of these 4 MBs will be early terminated, Comparator & MV Generator hardware calculates the MVs of the early terminated MBs, and then the search process starts for the next 4 MBs. If less than 4 MBs are early terminated, performance of the C-1BT ME hardware does not increase.

However, in this case, Control Unit stops the switching activity in the PE Arrays of the early terminated MBs in order to reduce power consumption of the C-1BT ME hardware. Switching activity in PEs, adder trees and comparators are reduced for early terminated MBs during the search process which takes 1127 clock cycles. As it is shown in Figure 5.5, PE Array used in Early Terminated C-1BT ME Hardware has additional multiplexers compared to the PE Array hardware shown in Figure 5.2. When a MB is early terminated, 0 is sent to PE Array for both the SW_T of 1 bit depth image and CM. Current MB values from CM and Current MB pixels from 1 bit depth image do not change during search process of a MB. Therefore, they are not multiplexed for reducing switching activity.
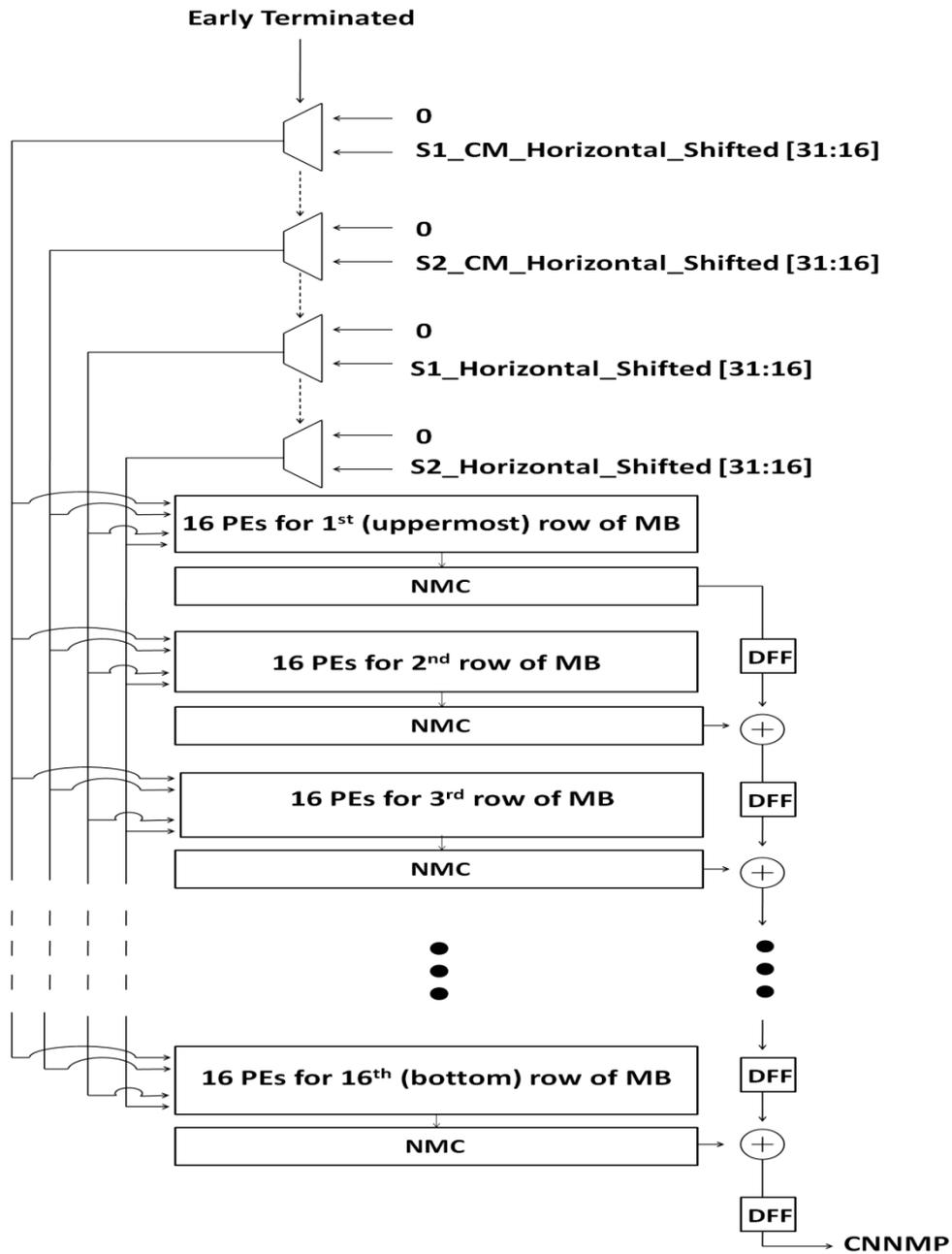
Figure 5.5 MB0 PE Array for Early Terminated C-1BT ME Hardware

Early Termination Decision hardware architecture is shown in Figure 5.6. Early Termination Decision hardware starts to work while CM values for the current MB are loaded from the off-chip memory to the PE Arrays. The number of "1"s in a row of CM for the current MB is calculated using NMC hardware. The NMC hardware architecture is shown in Figure 3.5 (b). Accumulators shown in Figure 5.6 calculate the total number of "1"s in CM for the current MB in 16 clock cycles.
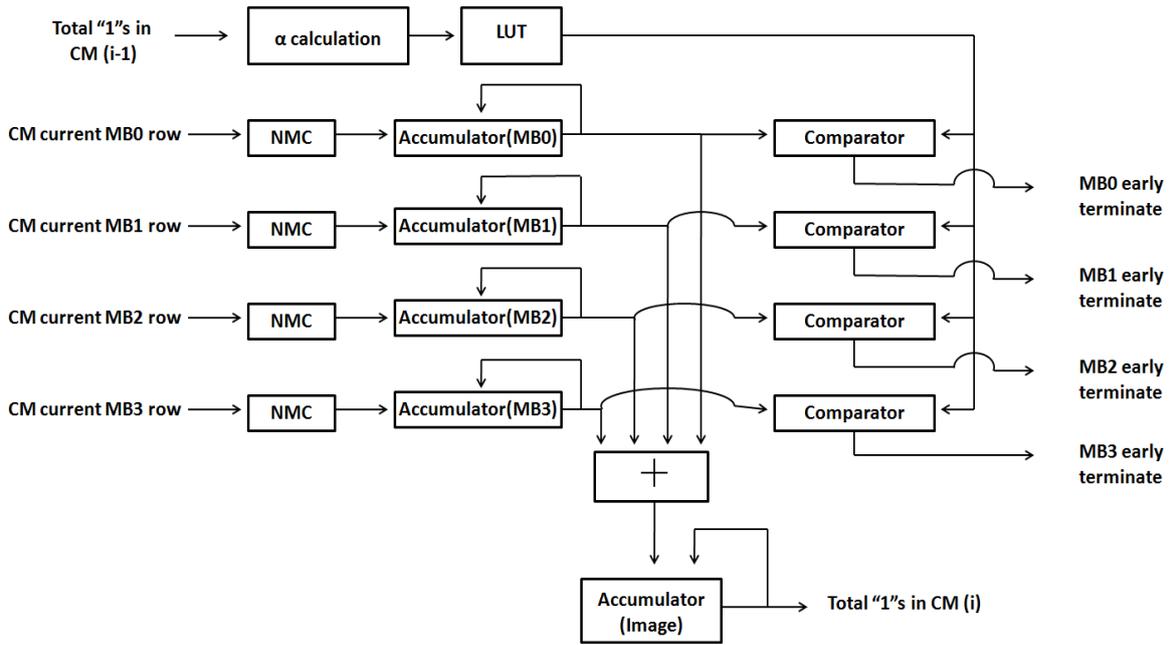
Figure 5.6 Early Termination Decision Hardware
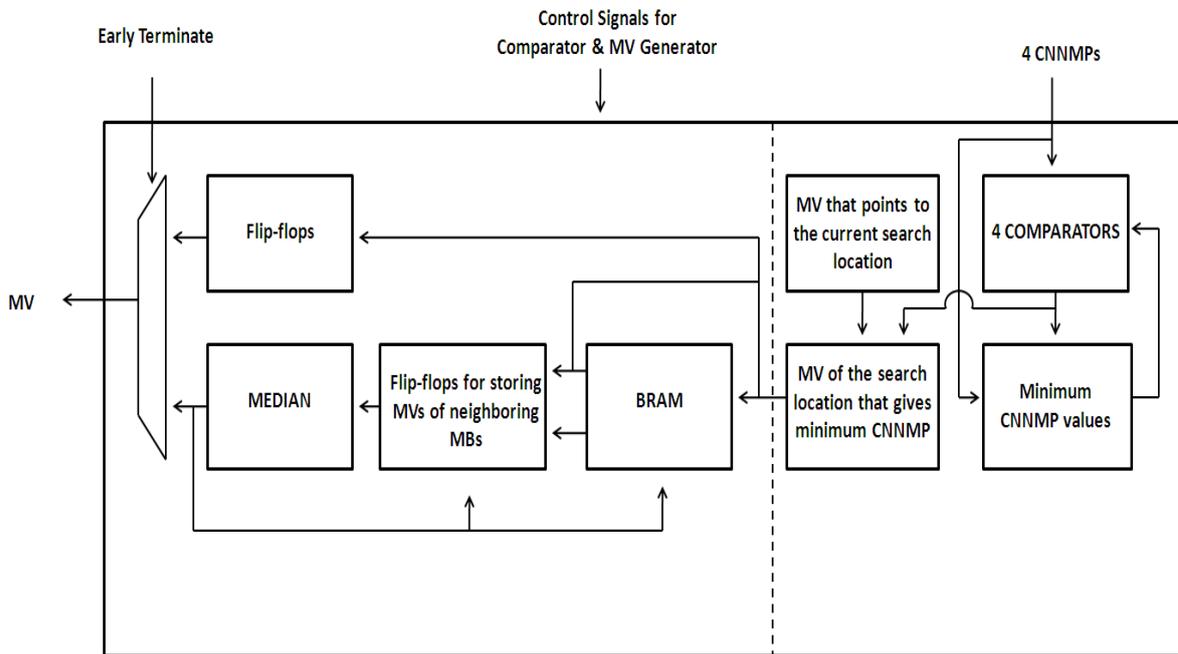
Figure 5.7 Comparator & MV Generator Hardware for Early Terminated C-1BT ME Hardware

As shown in Figure 5.6, Early Termination Decision hardware also calculates the total number of "1"s in CM of the current image. This value is used for calculating α for the next image. As it is shown in (2.8), calculating α requires division and multiplication operations. We implemented these division and multiplication operations by using addition and shift operations. After α is calculated, $256/\alpha$ is calculated. We implemented this calculation using LUTs. The output of the LUTs are compared with the final values in the accumulators. If total number of "1"s in CM of a MB is less than the output of LUTs, Early Termination Decision hardware decides that the search process will be skipped for this MB.

Comparator & MV Generator hardware architecture is shown in Figure 5.7. The left part of the dashed line is the hardware used for computing the MVs of the early terminated MBs. In C-1BT ME hardware, the MV pointing to the search location with the minimum CNNMP is taken as the MV of the corresponding MB. However, in Early Terminated C-1BT ME hardware, MVs of the left, upper-left and upper neighboring MBs are used to calculate the MVs of early terminated MBs. Therefore, the MVs of these neighboring MBs are stored in BRAMs. If a MB is early terminated, the multiplexer at the output of the Comparator & MV Generator selects the output of the median hardware as the MV of this MB.

5.3 Implementation Results

The proposed C-1BT ME and early terminated C-1BT ME hardware architectures are implemented in Verilog HDL. The Verilog RTL codes are synthesized with Mentor Graphics Precision RTL 2005b and mapped to a XC2VP30-7 FPGA using Xilinx ISE 8.2i. The hardware implementations are verified with post place & route simulations using Modelsim 6.1c.

The C-1BT ME hardware consumes 3838 slices (5211 LUTs), which is 28% of all the slices in Xilinx XC2VP30-7 FPGA. In addition, 4096 bits on-chip memory is used for storing SW_T pixels from the 1 bit depth image and 4096 bits on-chip memory is used for storing SW_T values from CM. Therefore, C-1BT ME hardware requires 8192 bits on-chip memory. These 8192 bits are stored in 64 addresses of 4 BRAMs.

Same as the FBS SRF ME hardware proposed in Chapter IV, 1127 clock cycles are required by the proposed C-1BT ME hardware for processing 4 MBs in a [-16, 16] search range. The proposed C-1BT ME hardware implementation can work at 190 MHz. Therefore, it is capable of processing 83 1920x1080 full HD frames per second.

Early Terminated C-1BT ME hardware consumes 4410 slices (6079 LUTs), which is 32% of all the slices in the same FPGA. This ME hardware stores MVs of the MBs in a row of the image in order to calculate the MVs of the early terminated MBs. Therefore, it uses 1440 bits additional on-chip memory. 1 BRAM is used for storing these 1440 bits. Therefore, the proposed Early Terminated C-1BT ME hardware requires 9630 bits on-chip memory. The proposed Early Terminated C-1BT ME hardware implementation can work at 189 MHz.

Early Termination Decision hardware works in parallel with the C-1BT ME hardware. The calculation of the MVs for early terminated MBs requires additional 4 clock cycles. However, the calculation of the MVs for early terminated MBs is done in parallel with the first 4 clock cycles of ME process for the next 4 MBs. Since the Early Termination Decision and Comparator & MV Generator hardware work in parallel with the C-1BT ME hardware, in the worst case, Early Terminated C-1BT ME hardware is capable of processing 83 1920x1080 full HD frames per second. When all of the 4 MBs are early terminated, it can process more than 83 1920x1080 full HD frames per second.

The comparison of the proposed C-1BT ME hardware and Early Terminated C-1BT ME hardware is shown in Table 5.1. Early Terminated C-1BT ME hardware uses 17% more LUTs and 13% more DFFs than C-1BT ME hardware on Xilinx XC2VP30-7 FPGA.

In order to compare the energy consumptions of the C-1BT ME hardware and the Early Terminated C-1BT ME hardware, we synthesized the Verilog RTL codes and mapped them to a XC5LX110FF676-3 FPGA using Xilinx ISE 11.4. We, then, estimated their power consumptions using Xilinx XPower Analyzer. In order to estimate the dynamic power consumption of a hardware implementation on a Xilinx Virtex 5 FPGA, timing simulation of the placed & routed netlist of that hardware implementation is done for one frame of ParkJoy HD video at 100 MHz using Mentor Graphics ModelSim 6.1c and the signal activities are

stored in a Value Change Dump (VCD) file. This VCD file is used for estimating the power consumption of that hardware implementation.

As it is shown in Table 5.2, Early Terminated C-1BT ME hardware requires more LUTs and DDFs than the C-1BT ME hardware. 4124 of the 8160 MBs in the full HD frame are early terminated, and 3560 of these 4124 MBs are 4 MBs (MB0, MB1, MB2 and MB3) processed in parallel. When less than 4 MBs processed in parallel are early terminated, the energy consumption of the Early Terminated C-1BT ME hardware reduces by the reduction in the switching activity in PE Arrays. When all 4 MBs processed in parallel are early terminated, both the energy consumption of the Early Terminated C-1BT ME hardware reduces and its speed increases. The results show that Early Terminated C-1BT ME hardware consumes 20% more dynamic power than C-1BT ME hardware. Because Early Termination Decision hardware and Comparator & MV Generator hardware works in parallel with C-1BT ME hardware. However, Early Terminated C-1BT ME hardware is 41% faster than the C-1BT ME hardware. Therefore, it has 26% less energy consumption than the C-1BT ME hardware.

Table 5.1 Comparison of Proposed Hardware Implementations

	C-1BT ME Hardware	**Early Terminated C-1BT ME Hardware**
Bit Depth	2	2
On-Chip Memory (bits)	8192	9630
Area	5211 LUTs 3572 DFFs	6079 LUTs 4049 DFFs
Maximum Frequency (MHz)	190	189
Search Range	[-16, 16]	[-16, 16]
Search Location / MB	1089	1089
Performance (1920x1080 fps)	83	> 83

Table 5.2 Comparison of Proposed Hardware Implementations on Virtex 5 FPGA

	C-1BT ME Hardware	Early Terminated C-1BT ME Hardware
Bit Depth	2	2
On-Chip Memory (bits)	8192	9630
Area	4529 LUTs 3667 DFFs	5035 LUTs 4100 DFFs
Maximum Frequency (MHz)	270	265
Search Range	[-16, 16]	[-16, 16]
Search Location / MB	1089	1089
Percentage of Early Terminated MBs	—	50.54%
Percentage of 4 MBs Processed in Parallel Early Terminated	—	86.32%
Dynamic Power Consumption (mW)	160	202
Time (ms)	26.21	15.22
Energy Consumption (mJ)	4.202	3.076
Energy Reduction	—	26.8%

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

In this thesis, first, we proposed high performance 1BT based ME hardware architectures with rectangular MB organization. The FBS SRF ME and VBS SRF ME hardware architectures using rectangular MB organization are faster and use less on-chip memory than the previous 1BT based ME hardware architectures in literature. Both the FBS SRF ME and VBS SRF ME hardware using rectangular MB organization are capable of processing 49 1920x1080 full HD frames per second.

Then, we proposed high performance 1BT based ME hardware architectures with square MB organization. The FBS SRF ME and VBS SRF ME hardware architectures using square MB organization are faster, use less logic area and less on-chip memory than the proposed FBS SRF ME and VBS SRF ME hardware architectures using rectangular MB organization. The proposed reconfigurable MRF ME hardware is the first 1BT based MRF ME hardware in the literature. In the proposed MRF ME hardware, the number and selection of RFs can be statically configured based on the application requirements in order to trade-off ME performance and computational complexity. All the FBS SRF ME, VBS SRF ME and MRF ME hardware using square MB organization are capable of processing 83 1920x1080 full HD frames per second.

Finally, we proposed high performance C-1BT ME and early terminated C-1BT ME hardware architectures. Early terminated C-1BT ME hardware is 41% faster and consumes 26% less energy than C-1BT ME hardware.

As future work, 1BT based VBS MRF ME hardware can be designed by integrating FBS MRF ME and VBS SRF ME hardware architectures. A 1BT hardware can be implemented and integrated with the proposed ME hardware architectures. The search range of the proposed hardware architectures can be increased for better ME performance. ASIC implementation results of the proposed ME hardware architectures can be presented.

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 Electron.*, vol. 54, no.2, pp. 863-869, May 2008.

[4] C. Wei, H. Hui, T. Jiarong, and M. Hao, "A High-performance Reconfigurable VLSI Architecture for VBSME inH.264," *IEEE Trans on Consumer Electron.*, vol. 54, no. 3, pp. 1338-1345, Aug. 2008.

[5] T. Moorthy, and A. Ye, "A Scalable Computing and Memory Architecture for Variable Block Size Motion Estimation on Field-Programmable Gate Arrays," *International Conference on Field Programmable Logic*, pp. 83-88, Sept. 2008.

[6] K. Lee, G. Jeon, and J. Jeong, "Fast Reference Frame Selection Algorithm for H.264/AVC," *IEEE Trans. on Consumer Electron.*, vol. 55, pp. 773–779, May 2009.

[7] L. Shen, Z. Liu, Z. Zhang, and G. Wang, "An Adaptive and Fast Multiframe Selection Algorithm for H.264 Video Coding," *IEEE Signal Process. Letters*, vol. 14, no. 11, pp. 836-839, Nov. 2007.

[8] 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.

[9] 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.

[10] 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.

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

[12] W. M. Chao, C. W. Hsu, Y. C. Chang, and L. G. Chen, "A Novel Motion Estimator Supporting Diamond Search and Fast Full Search," *IEEE ISCAS*, May 2002.

[13] O. Tasdizen, A. Akin, H. Kukner, and I. Hamzaoglu, "Dynamically Variable Step Search Motion Estimation Algorithm and a Dynamically Reconfigurable Hardware for Its Implementation," *IEEE Trans. on Consumer Electron.*, vol. 55, no. 3, Aug 2009.

[14] 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.

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

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

[17] 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.

[18] 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.

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

[20] 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 Electron.*, vol. 54, no. 4, Nov. 2008.

[21] A. Bahari, T. Arslan, and A. T. Erdogan, "Low-Power H.264 Video Compression Architectures for Mobile Communication", *IEEE Trans. on Circuits Syst. Video Technol.,* vol. 19, no. 9, Sept. 2009.

[22] A. Akin, Y. Dogan, and I. Hamzaoglu, "A High Performance Hardware Architecture for One Bit Transform Based Motion Estimation Algorithms," *Euromicro Conference on DSD*, Patras, Greece, Aug. 2009.

[23] A. Akin, Y. Dogan, and I. Hamzaoglu, "High Performance Hardware Architectures for One Bit Transform Based Motion Estimation Algorithms," *IEEE Trans. on Consumer Electron.*, vol. 55, pp. 773–779, May 2009.

[24] A. Akin, G. Sayilar, I. Hamzaoglu, "High Performance Hardware Architectures for One Bit Transform Based Single and Multiple Reference Frame Motion Estimation," *IEEE Trans. on Consumer Electron.*, May 2010.

[25] A. Akin, G. Sayilar, I. Hamzaoglu, "A Reconfigurable Hardware for One Bit Transform based Multiple Reference Frame Motion Estimation", *DATE Conference*, March 2010.

[26] G. Stewart, D. Renshaw, and M. Riley, "A Novel Motion Estimation Power Reduction Technique," *International Conference on Field Programmable Logic*, pp. 546–549, August 2007.

[27] S. Yalcin, H. F. Ates and I. Hamzaoglu, "A High Performance Hardware Architecture for an SAD Reuse based Hierarchical Motion Estimation Algorithm for H.264 Video Coding", *International Conference on Field Programmable Logic*, August 2005.

[28] S. Erturk, O. Urhan, and I. Hamzaoglu, "Final Report of TUBITAK Project 107E179," May 2010.