

LOW ENERGY HEVC VIDEO COMPRESSION HARDWARE DESIGNS

by
Ercan Kalalı

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

Sabancı University
August 2013

LOW ENERGY HEVC VIDEO COMPRESSION HARDWARE DESIGNS

APPROVED BY:

Assoc. Prof. Dr. İlker Hamzaoğlu
(Thesis Supervisor)

.....


Assoc. Prof. Dr. Müştak Erhan Yalçın

.....


Asst. Prof. Dr. Hakan Erdoğan

.....


DATE OF APPROVAL:

© Ercan Kalalı 2013
All Rights Reserved

To my Mother and Father

ACKNOWLEDGEMENT

I would like to thank my supervisor, Dr. İlker Hamzaoğlu for all his guidance, support, and patience throughout my MS study. I appreciate very much for his suggestions, detailed reviews, invaluable advices and life lessons. I particularly want to thank him for his confidence and belief in me during my study. It has been a great honor for me to work under his guidance.

I like to convey my heartiest thanks to Yusuf Adıbelli for his unlimited support and encouragement.

I would like to thank to all members of System-on-Chip Design and Testing Lab, Erdem Özcan, Kamil Erdayandı, Yusuf Akşehir, Zafer Özcan, Serkan Yalman, and Hasan Azgın who have been greatly supportive during my study.

Special thanks to my family. This thesis is dedicated with love and gratitude to my parents for their constant support and encouragement for going through my tough periods with me.

Finally, I would like to acknowledge Sabancı University and Scientific and Technological Research Council of Turkey (TUBITAK) for supporting me throughout my graduate education.

LOW ENERGY HEVC VIDEO COMPRESSION HARDWARE DESIGNS

Ercan Kalalı

Electronics, MS Thesis, 2013

Thesis Supervisor: Assoc. Prof. İlker HAMZAOĞLU

Keywords: HEVC, Intra Prediction, Sub-Pixel Interpolation

ABSTRACT

Joint collaborative team on video coding (JCT-VC) recently developed a new international video compression standard called High Efficiency Video Coding (HEVC). HEVC has 37% better compression efficiency than H.264 which is the current state-of-the-art video compression standard. HEVC achieves this video compression efficiency by significantly increasing the computational complexity. Therefore, in this thesis, we propose novel computational complexity and energy reduction techniques for intra prediction algorithm used in HEVC video encoder and decoder. We quantified the computation reductions achieved by these techniques using HEVC HM reference software encoder. We designed efficient hardware architectures for these video compression algorithms used in HEVC. We also designed a reconfigurable sub-pixel interpolation hardware for both HEVC encoder and decoder. We implemented these hardware architectures in Verilog HDL. We mapped the Verilog RTL codes to a Xilinx Virtex 6 FPGA and estimated their power consumptions on this FPGA using Xilinx XPower Analyzer tool. The proposed techniques significantly reduced the energy consumptions of these FPGA implementations in some cases with no PSNR loss and in some cases with very small PSNR loss.

DÜŞÜK ENERJİLİ HEVC VIDEO SIKIŞTIRMA DONANIM TASARIMLARI

Ercan Kalalı

Elektronik Müh., Yüksek Lisans Tezi, 2013

Tez Danışmanı: Doç. Dr. İlker HAMZAOĞLU

Anahtar Kelimeler: HEVC, Çerçeve İçi Öngörü, Ara Piksel Hesaplama

ÖZET

Joint Collaborative Team on Video Coding (JCT-VC) yüksek verimli video kodlama (HEVC) isminde yeni bir video sıkıştırma standardı geliştirdi. HEVC günümüzde kullanılan H.264 standardına göre 37% daha iyi performans sağlıyor. HEVC bu video sıkıştırma verimini hesaplama karmaşıklığını önemli ölçüde artırarak başarıyor. Bu nedenle, bu tezde HEVC video kodlayıcı ve kod çözücü için kullanılan çerçeve içi öngörü algoritmaları için yeni hesaplama karmaşıklığı ve enerji azaltma teknikleri önerildi. Önerilen tekniklerin hesaplama miktarında yaptığı azalma HEVC referans yazılımı (HM) kullanılarak belirlendi. Bu HEVC video sıkıştırma algoritmaları için verimli donanım mimarileri tasarlandı. Ayrıca HEVC video kodlayıcı ve kod çözücü ara pikselleri oluşturma algoritmasının yeniden yapılandırılabilir donanım tasarımı yapıldı. Bu donanım mimarileri Verilog donanım tasarlama dili ile gerçekleştirildi. Verilog HDL kodları Xilinx Virtex 6 FPGA'ine sentezlendi ve Xilinx XPower Analyzer ile bu FPGA'deki güç tüketimi tahmini yapıldı. Önerilen teknikler bu FPGA gerçeklemelerinin enerji tüketimini bazen hiçbir PSNR kaybı olmaksızın, bazen de çok küçük PSNR kaybı ile önemli miktarda azaltmıştır.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	V
1 ABSTRACT	VI
2 ÖZET	VII
3 TABLE OF CONTENTS	VIII
LIST OF FIGURES	X
LIST OF TABLES	XI
LIST OF ABBREVIATIONS	XII
1 CHAPTER I INTRODUCTION	1
1.1 HEVC Video Compression Standard	1
1.2 Thesis Contributions	3
1.3 Thesis Organization	5
2 CHAPTER II A LOW ENERGY INTRA PREDICTION HARDWARE FOR HIGH EFFICIENCY VIDEO CODING	6
2.1 HEVC Intra Prediction Algorithm	8
2.2 Proposed Computation Reduction Techniques	11
2.3 Proposed HEVC Intra Prediction Hardware	17
2.4 HEVC Intra Prediction Hardware Implementation on FPGA Board	22
3 CHAPTER III A HIGH PERFORMANCE AND LOW ENERGY INTRA PREDICTION HARDWARE FOR HEVC VIDEO DECODER	24
3.1 Proposed Computation Reduction Techniques	26
3.2 Proposed Intra Prediction Hardware Architecture and Its Energy Consumption	31

4	CHAPTER IV A RECONFIGURABLE HEVC SUB-PIXEL INTERPOLATION HARDWARE.....	35
4.1	HEVC Sub-Pixel Interpolation Algorithm.....	36
4.2	Proposed Reconfigurable HEVC Sub-Pixel Interpolation Hardware	38
5	CHAPTER V CONCLUSIONS AND FUTURE WORK.....	43
6	BIBLIOGRAPHY	44

LIST OF FIGURES

Figure 1.1 HEVC Encoder Block Diagram.....	2
Figure 1.2 HEVC Decoder Block Diagram.....	2
Figure 2.1 HEVC Intra Prediction Mode Directions.....	9
Figure 2.2 Prediction Equations for 8x8 Luma Prediction Mode.....	10
Figure 2.3 Neighboring Pixels of 4x4 and 8x8 PUs.....	11
Figure 2.4 Rate Distortion Curves of Original 4x4 and 8x8 Intra Prediction Algorithms and 4x4 and 8x8 Intra Prediction Algorithms with PSCR Technique for 4bT	16
Figure 2.5 HEVC Intra Prediction Hardware.....	18
Figure 2.6 HEVC Intra Prediction Datapath	20
Figure 2.7 HEVC Intra Prediction Hardware FPGA Board Implementation.....	23
Figure 3.1 PU Sizes Selected by HEVC Video Encoder for Intra Prediction (QP = 27).....	28
Figure 3.2 Prediction Modes Selected by HEVC Video Encoder for Intra Prediction	28
Figure 3.3 Intra Prediction Hardware for HEVC Video Decoding	31
Figure 3.4 Intra Prediction Datapath	32
Figure 4.1 Integer, Half and Quarter Pixels	37
Figure 4.2 Original Sub-Pixel Interpolation Datapath	38
Figure 4.3 HEVC Sub-Pixel Interpolation Hardware.....	39
Figure 4.4 Reconfigurable Sub-Pixel Interpolation Datapath.....	40

LIST OF TABLES

Table 2.1 Some of The HEVC Intra Prediction Equations	12
Table 2.2 Computation Reductions by Date Reuse.....	13
Table 2.3 Percentages of 8x8 PUs with Equal and Similar Pixels for 1920x1080 Video Frames	14
Table 2.4 Percentages of 8x8 PUs with Equal and Similar Pixels for 1280x720 Video Frames	15
Table 2.5 Computation Reductions by PECR After Data Reuse	15
Table 2.6 Computation Reductions by PSCR After Data Reuse	16
Table 2.7 Average PSNR Comparison of PSCR Technique.....	17
Table 2.8 Energy Consumption Reduction for 1920x1080 Video Frames	21
Table 2.9 Energy Consumption Reduction for 1280x720 Video Frames	21
Table 3.1 Some of The HEVC Intra Prediction Equations	27
Table 3.2 Computation Reductions by Data Reuse for 1920 x 1080 Frames	29
Table 3.3 Computation Reductions by Data Reuse for 1280 x 720 Frames	29
Table 3.4 Percentages of 8x8 PUs with Equal Pixels.....	29
Table 3.5 Computation Reductions (%) by PECR After Data Reuse	30
Table 3.6 Comparison Overhead.....	30
Table 3.7 Energy Consumption Reduction for 1280x720 Video Frames	34
Table 3.8 Energy Consumption Reduction for 1920x1080 Video Frames	34
Table 4.1 Necessary Sub-Pixels for Possible X Fraction and Y Fraction Values.....	37
Table 4.2 Amounts of Computations for Sub-Pixel Interpolation	37
Table 4.3 FIR Filter Coefficients	40
Table 4.4 Power Consumption Reductions for 1920x1080 Video Frames.....	42

LIST OF ABBREVIATIONS

ALF	Adaptive Loop Filter
BRAM	Block Ram
CABAC	Context Adaptive Binary Arithmetic Coding
CU	Coding Unit
DBF	Deblocking Filter
DCT	Discrete Cosine Transform
DST	Discrete Sine Transform
DVI	Digital Visual Interface
FPGA	Field Programmable Gate Array
HD	High Definition
HEVC	High Efficiency Video Coding
HM	HEVC Test Model
PSNR	Peak Signal to Noise Ratio
PU	Prediction Unit
QP	Quantization Parameter
SAO	Sample Adaptive Offset
TU	Transform Unit
UART	Universal Asynchronous Receiver/Transmitter
VCD	Value Change Dump

CHAPTER I

INTRODUCTION

1.1 HEVC Video Compression Standard

Since better coding efficiency is required for high resolution videos, Joint Collaborative Team on Video Coding (JCT-VC) recently developed a new video compression standard called High Efficiency Video Coding (HEVC) [1, 2, 3]. HEVC provides 37% better coding efficiency than H.264 which is the current state-of-the-art video compression standard. HEVC also provides 23% bit rate reduction for the intra prediction only case [4]. The video compression efficiency achieved in HEVC standard is not a result of any single feature but rather a combination of a number of encoding tools such as intra prediction, motion estimation, deblocking filter, sample adaptive offset (SAO) and entropy coder.

The top-level block diagram of an HEVC encoder and decoder are shown in Figure 1.1 and Figure 1.2, respectively. An HEVC encoder has a forward path and a reconstruction path. The forward path is used to encode a video frame by using intra and inter predictions and to create the bit stream after the transform and quantization process. Reconstruction path in the encoder ensures that both encoder and decoder use identical reference frames for intra and inter prediction because a decoder never gets original images.

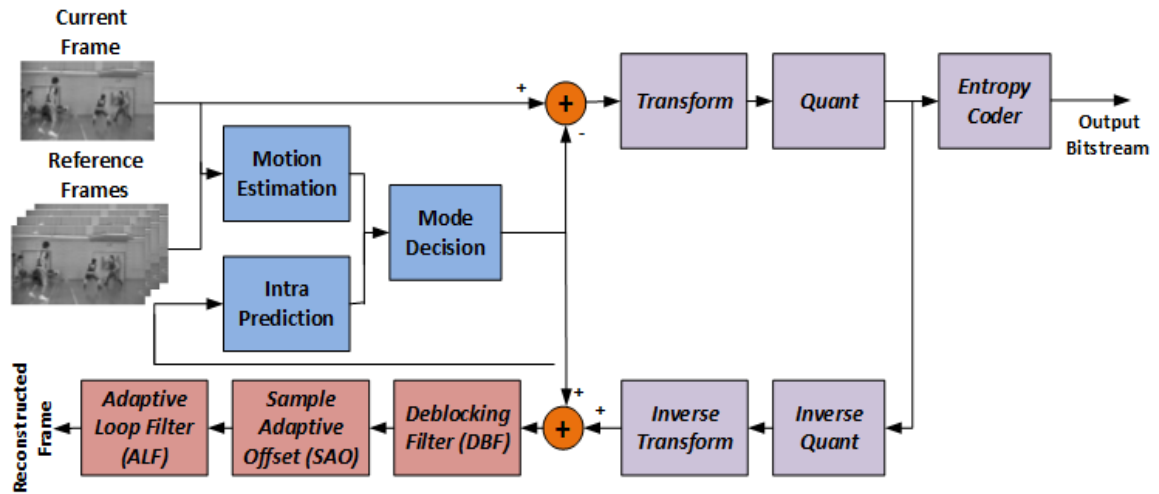


Figure 1.1 HEVC Encoder Block Diagram

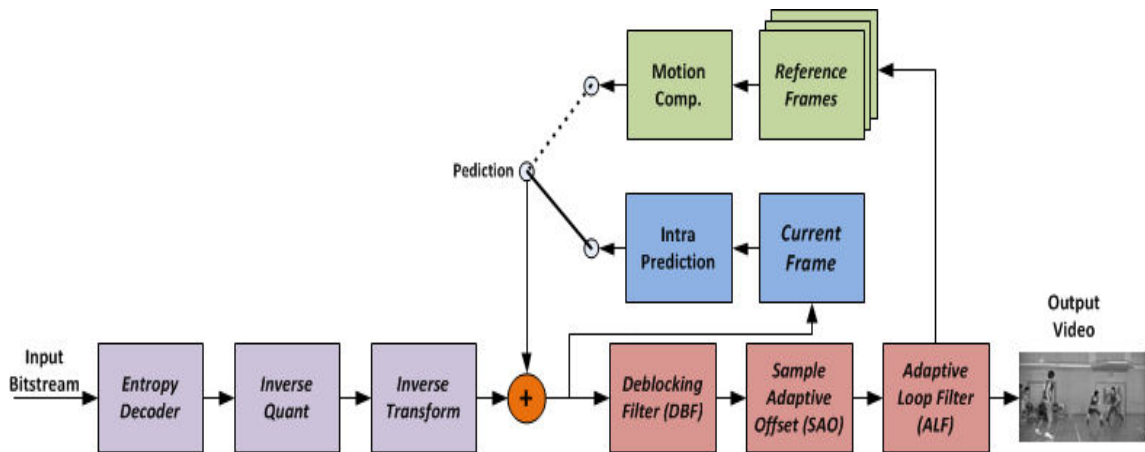


Figure 1.2 HEVC Decoder Block Diagram

In the forward path, frame is divided into coding units (CU) that can be an 8x8, 16x16, 32x32 or 64x64 pixel block. Each CU is encoded in intra or inter mode depending on the mode decision. Intra and inter prediction processes use prediction unit (PU) partitioning inside the CUs. Prediction unit (PU) sizes can be from 4x4 up to 64x64. Mode decision determines whether a PU will be coded intra or inter mode based on video quality and bit-rate. After mode decision determines the prediction mode, predicted block is subtracted from original block, and residual data is generated. Then, residual data transformed by discrete cosine transform (DCT) and quantized. Transform unit (TU) sizes can be from 4x4 up to 32x32. Finally, entropy coder generates the encoded bitstream.

Reconstruction path begins with inverse quantization and inverse transform operations. The quantized transform coefficients are inverse quantized and inverse transformed to generate the reconstructed residual data. Since quantization is a lossy process, inverse quantized and inverse transformed coefficients are not identical to the original residual data. The reconstructed residual data are added to the predicted pixels in order to create the reconstructed frame. DBF is, then, applied to reduce the effects of blocking artifacts in the reconstructed frame.

H.264 and HEVC intra prediction algorithms predict the pixels of a block from the pixels of its already coded and reconstructed neighboring blocks. In H.264, there are 9 intra prediction modes for 4x4 luminance blocks, and 4 intra prediction modes for 16x16 luminance blocks [5, 6]. In HEVC, for the luminance component of a frame, intra PU sizes can be from 4x4 up to 64x64 and number of intra prediction modes for a PU can be up to 35 [1, 7].

In order to increase the performance of integer pixel motion estimation, sub-pixel (half and quarter) accurate variable block size motion estimation is performed in H.264 and HEVC. In H.264, a 6-tap FIR filter is used for half-pixel interpolation and a bilinear filter is used for quarter-pixel interpolation [6]. In HEVC, 3 different 8-tap FIR filters are used for both half-pixel and quarter-pixel interpolations [1, 2, 32].

Integer based DCT is used in HEVC same as H.264. In H.264, transformation block sizes can be 4x4 or 8x8. In HEVC, TU sizes can be from 4x4 up to 32x32. In addition to DCT, HEVC uses discrete sine transform (DST) for the 4x4 intra prediction case [1, 2, 24].

Entropy coder uses the context adaptive binary arithmetic coding (CABAC) similar to H.264 with several improvements [2].

Deblocking filter algorithm reduces the blocking artifacts on the edges of prediction units. Sample adaptive offset (SAO) and adaptive loop filter (ALF) are added to deblocking filter process in HEVC which are not used in previous video compression standards [1, 2].

1.2 Thesis Contributions

We propose using data reuse technique for HEVC intra prediction algorithm. In HEVC, intra luminance angular prediction modes have identical equations. There are identical equations between 4x4 and 8x8 luminance angular prediction modes as well.

Therefore, we propose calculating the common prediction equations for all 4x4 and 8x8 luminance angular prediction modes only once and using the results for the corresponding prediction modes. In this way, the amount of computations performed by HEVC intra prediction algorithm is reduced up to 84%.

We propose pixel equality based computation reduction (PECR) technique for reducing the amount of computations performed by HEVC intra prediction algorithm and therefore reducing the power consumption of HEVC intra prediction hardware significantly without any PSNR and bit rate loss. The proposed technique performs a small number of comparisons among neighboring pixels of the current PU before the intra prediction process. If the pixels used in a prediction equation are equal, the predicted pixel by this equation is equal to these pixels. Therefore, this prediction equation simplifies to a constant value and prediction calculation for this equation becomes unnecessary. In this way, the amount of computations performed by HEVC intra prediction algorithm is reduced up to 65% without any PSNR and bitrate loss.

We also propose using pixel similarity based computation reduction (PSCR) technique for HEVC intra prediction algorithm as well. PSCR technique compares the pixels used in the prediction equations of angular intra prediction modes. If the pixels used in a prediction equation are similar, the predicted pixel by this equation is assumed to be equal to one of these pixels. Therefore, this prediction equation simplifies to a constant value and prediction calculation for this equation becomes unnecessary. In this way, the amount of computations performed by HEVC intra prediction algorithm is reduced up to 92% with a small PSNR loss.

We also implemented an efficient 4x4 and 8x8 intra luminance angular prediction hardware including the proposed techniques using Verilog HDL. We quantified the impact of the proposed techniques on the power consumption of this hardware on a Xilinx Virtex 6 FPGA using Xilinx XPower. PECR and PSCR techniques reduced the energy consumption of this hardware up to 40% and 66% [11].

Since intra prediction algorithm used in HEVC decoder has to find the intra prediction only for the prediction mode selected by HEVC encoder, in this thesis, we adapt the data reuse technique for HEVC decoder, and we propose calculating the common prediction equations for each 4x4 and 8x8 luminance prediction mode only once and using the results for the corresponding prediction mode. We also use the PECR technique for the intra prediction algorithm in the HEVC decoder.

We also designed a high performance intra prediction hardware for angular prediction modes of 4x4 and 8x8 PU sizes including the proposed techniques for HEVC video decoding. The proposed hardware is implemented using Verilog HDL. We quantified the impact of PECR technique on the energy consumption of the proposed intra prediction hardware for HEVC video decoding including data reuse technique on this FPGA using Xilinx XPower Analyzer tool, and PECR technique reduced its energy consumption up to 42% [16].

We designed a reconfigurable HEVC sub-pixel (half-pixel and quarter-pixel) interpolation hardware for all PU sizes. The proposed hardware is implemented using Verilog HDL. The proposed reconfigurability reduces the area and power consumption of HEVC sub-pixel interpolation hardware more than 30%. The proposed hardware, in the worst case, can process 64 quad full HD (2560x1600) video frames per second [32].

1.3 Thesis Organization

The rest of the thesis is organized as follows.

Chapter II explains HEVC intra prediction algorithm. It presents the proposed Data Reuse, PECR and PSCR techniques for HEVC intra prediction. It describes the proposed low energy HEVC intra prediction hardware including these techniques and presents its implementation results.

Chapter III presents the data reuse and PECR techniques for intra prediction in HEVC video decoder. It describes the proposed low energy and high performance intra prediction hardware for HEVC video decoding including these techniques and presents its implementation results.

Chapter IV explains HEVC sub-pixel interpolation algorithm. It describes the proposed reconfigurable HEVC sub-pixel interpolation hardware and presents its implementation results.

Chapter V presents conclusions and future work.

CHAPTER II

A LOW ENERGY INTRA PREDICTION HARDWARE FOR HIGH EFFICIENCY VIDEO CODING

Joint collaborative team on video coding (JCT-VC) recently developed a new video compression standard called High Efficiency Video Coding (HEVC) [1, 2, 3]. HEVC provides 37% better coding efficiency than H.264 which is the current state-of-the-art video compression standard. HEVC also provides 23% bit rate reduction for the intra prediction only case [4].

Intra prediction algorithm predicts the pixels of a block from the pixels of its already coded and reconstructed neighboring blocks. In H.264, there are 9 intra prediction modes for 4x4 luminance blocks, and 4 intra prediction modes for 16x16 luminance blocks [5, 6]. In HEVC, for the luminance component of a frame, intra prediction unit (PU) sizes can be from 4x4 up to 64x64 and number of intra prediction modes for a PU can be up to 35 [1, 7].

Pixel equality and pixel similarity based techniques, and data reuse technique are proposed for reducing amount of computations performed by H.264 intra prediction algorithm in [8, 9, 10]. Since HEVC intra prediction algorithm requires significantly more computations than H.264 intra prediction algorithm, in this thesis, we propose pixel equality and pixel similarity based techniques, and data reuse technique for reducing amount of computations performed by HEVC intra prediction algorithm, and therefore reducing energy consumption of HEVC intra prediction hardware.

We propose using data reuse technique for HEVC intra prediction algorithm. In

HEVC, intra 4x4 and 8x8 luminance angular prediction modes have identical equations. There are identical equations between 4x4 and 8x8 luminance angular prediction modes as well. Therefore, we propose calculating the common prediction equations for all 4x4 and 8x8 luminance angular prediction modes only once and using the results for the corresponding prediction modes.

We proposed using pixel equality based computation reduction (PECR) technique for HEVC intra prediction algorithm [11]. PECR technique compares the pixels used in the prediction equations of angular intra prediction modes. If the pixels used in a prediction equation are equal, the predicted pixel by this equation is equal to these pixels. Therefore, this prediction equation simplifies to a constant value and prediction calculation for this equation becomes unnecessary. In this way, the amount of computations performed by HEVC intra prediction algorithm is reduced significantly without any PSNR and bitrate loss.

We propose using pixel similarity based computation reduction (PSCR) technique for HEVC intra prediction algorithm as well. PSCR technique compares the pixels used in the prediction equations of angular intra prediction modes. If the pixels used in a prediction equation are similar, the predicted pixel by this equation is assumed to be equal to one of these pixels. Therefore, this prediction equation simplifies to a constant value and prediction calculation for this equation becomes unnecessary. In this way, the amount of computations performed by HEVC intra prediction algorithm is reduced even further with a small PSNR loss.

The simulation results obtained by HEVC Test Model HM 5.2 encoder software [12] for several benchmark videos showed that data reuse technique achieved up to 84% computation reduction. PECR technique after data reuse achieved up to 65% computation reduction, and PSCR technique after data reuse achieved up to 93% computation reduction with a small comparison overhead.

We designed a low energy HEVC intra prediction hardware for angular prediction modes of 4x4 and 8x8 PU sizes including PECR technique [11]. We also included PSCR technique in this hardware. Because, 94% of intra prediction uses 4x4 and 8x8 PU sizes [13]. The proposed hardware is implemented using Verilog HDL. The Verilog RTL code is mapped to a Xilinx Virtex 6 FPGA, and it is verified to work at 150 MHz by post place & route simulations. The FPGA implementation is also verified to work correctly on a Xilinx Virtex 6 FPGA board. The proposed FPGA implementation can process 30 full HD (1920x1080) video frames per second. We

quantified the impact of PECR and PSCR techniques on the energy consumption of the proposed HEVC intra prediction hardware including data reuse technique on this FPGA using Xilinx XPower tool. PECR and PSCR techniques reduced the energy consumption of this hardware on this FPGA up to 40% and 66%, respectively.

A HEVC intra prediction hardware only for 4x4 PU size is presented in [13]. However, no power reduction technique is used in this hardware, and its power consumption is not reported.

2.1 HEVC Intra Prediction Algorithm

HEVC intra prediction algorithm predicts the pixels in prediction units (PU) of a coding unit (CU), which is similar to macroblock in H.264, using the pixels in the available neighboring PUs. For the luminance component of a frame, 4x4, 8x8, 16x16, 32x32 and 64x64 PU sizes are available.

There are 16 angular prediction modes for 4x4 PU size, 33 angular prediction modes for 8x8, 16x16 and 32x32 PU sizes, and 2 angular prediction modes for 64x64 PU size. In addition to angular prediction modes shown in Figure 2.1, there are DC and planar prediction modes for all PU sizes [1].

In HEVC intra prediction algorithm, first, reference main array is determined. If the prediction mode is equal to or greater than 18, reference main array is selected from above neighboring pixels. However, first four pixels of this array are reserved to left neighboring pixels, and if prediction angle is less than zero, these pixels are assigned to the array. If the prediction mode is less than 18, reference main array is selected from left neighboring pixels. However, first four pixels of this array are reserved to above neighboring pixels, and if prediction angle is less than zero, these pixels are assigned to the array [1].

After the reference main array is determined, the index to this array and the coefficient of pixels are calculated as shown in Equation (1.1) and (1.2), respectively.

$$iIdx = ((y+1)*intraPredAngle) \gg 5 \quad (1.1)$$

$$iFact = ((y+1)*intraPredAngle) \& 31 \quad (1.2)$$

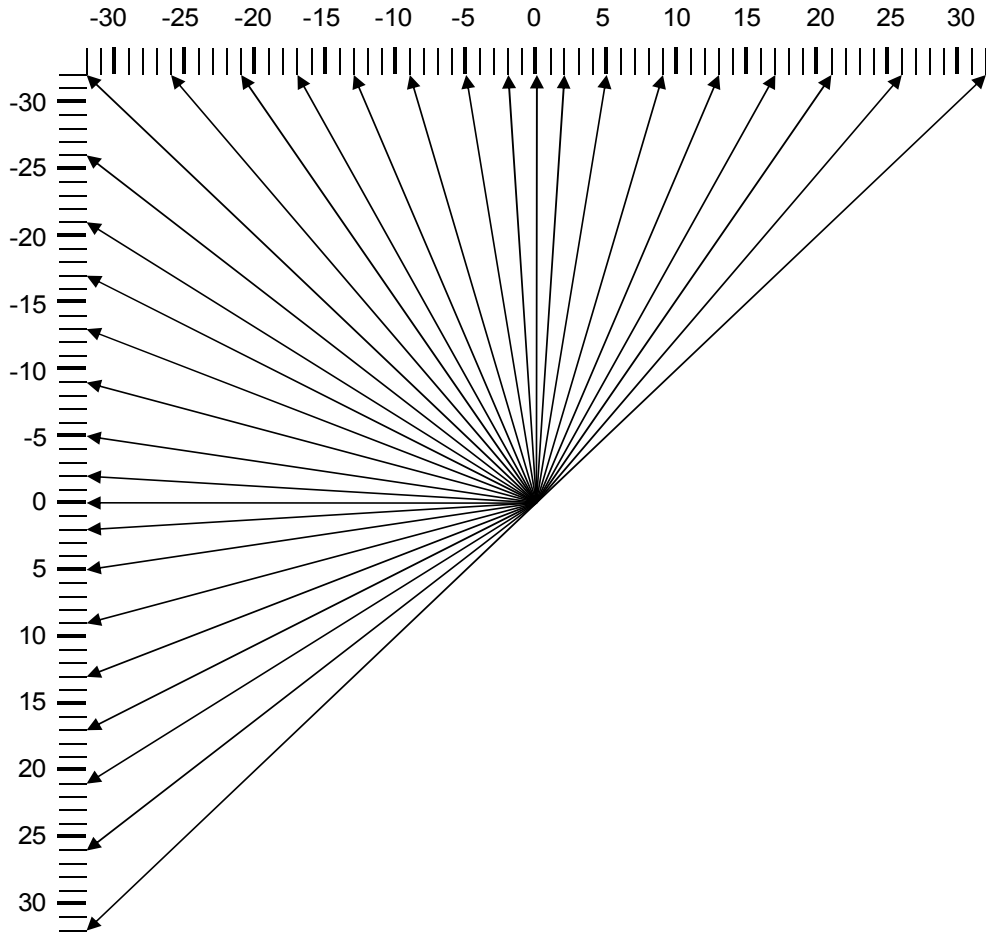


Figure 2.1 HEVC Intra Prediction Mode Directions

If $iFact$ is equal to 0, neighboring pixels are copied directly to predicted pixels. Otherwise, predicted pixels are calculated as shown in Equation (1.3).

$$\text{predSamples}[x,y] = ((32-iFact)*\text{refMain}[x+iIdx+1] + iFact*\text{refMain}[x+iIdx+2] + 16) \gg 5 \quad (1.3)$$

The reference main array and prediction equations for the 8x8 intra prediction mode 8 with prediction angle 5 are shown in Figure 2.2.

$$\text{refmain} = [0, 0, 0, 0, 0, 0, 0, 0, R, I, J, K, L, M, N, O, \\ P, HI, HJ, HK, HL, HM, HN, HO, HP]$$

$\text{pred}[0,0] = \text{pred}[1,0] =$
 $\text{pred}[2,0] = \text{pred}[3,0] =$
 $\text{pred}[4,0] = \text{pred}[5,0] = [27*I + 5*J + 16] \ggg 5$
 $\text{pred}[6,0] = \text{pred}[7,0] = [27*J + 5*K + 16] \ggg 5$
 $\text{pred}[0,1] = \text{pred}[1,1] =$
 $\text{pred}[2,1] = \text{pred}[3,1] =$
 $\text{pred}[4,1] = \text{pred}[5,1] = [22*J + 10*K + 16] \ggg 5$
 $\text{pred}[6,1] = \text{pred}[7,1] = [22*K + 10*L + 16] \ggg 5$
 $\text{pred}[0,2] = \text{pred}[1,2] =$
 $\text{pred}[2,2] = \text{pred}[3,2] =$
 $\text{pred}[4,2] = \text{pred}[5,2] = [17*K + 15*L + 16] \ggg 5$
 $\text{pred}[6,2] = \text{pred}[7,2] = [17*L + 15*M + 16] \ggg 5$
 $\text{pred}[0,3] = \text{pred}[1,3] =$
 $\text{pred}[2,3] = \text{pred}[3,3] =$
 $\text{pred}[4,3] = \text{pred}[5,3] = [12*L + 20*M + 16] \ggg 5$
 $\text{pred}[6,3] = \text{pred}[7,3] = [12*M + 20*N + 16] \ggg 5$
 $\text{pred}[0,4] = \text{pred}[1,4] =$
 $\text{pred}[2,4] = \text{pred}[3,4] =$
 $\text{pred}[4,4] = \text{pred}[5,4] = [7*M + 25*N + 16] \ggg 5$
 $\text{pred}[6,4] = \text{pred}[7,4] = [7*N + 25*O + 16] \ggg 5$
 $\text{pred}[0,5] = \text{pred}[1,5] =$
 $\text{pred}[2,5] = \text{pred}[3,5] =$
 $\text{pred}[4,5] = \text{pred}[5,5] = [2*N + 30*O + 16] \ggg 5$
 $\text{pred}[6,5] = \text{pred}[7,5] = [2*O + 30*P + 16] \ggg 5$
 $\text{pred}[0,6] = \text{pred}[1,6] =$
 $\text{pred}[2,6] = \text{pred}[3,6] =$
 $\text{pred}[4,6] = \text{pred}[5,6] = [29*O + 3*P + 16] \ggg 5$
 $\text{pred}[6,6] = \text{pred}[7,6] = [29*P + 3*VI + 16] \ggg 5$
 $\text{pred}[0,7] = \text{pred}[1,7] =$
 $\text{pred}[2,7] = \text{pred}[3,7] =$
 $\text{pred}[4,7] = \text{pred}[5,7] = [24*P + 8*VI + 16] \ggg 5$
 $\text{pred}[6,7] = \text{pred}[7,7] = [24*VI + 8*VJ + 16] \ggg 5$

Figure 2.2 Prediction Equations for 8x8 Luma Prediction Mode

2.2 Proposed Computation Reduction Techniques

In HEVC, intra 4x4 and 8x8 luminance angular prediction modes have identical equations. There are identical equations between 4x4 and 8x8 luminance angular prediction modes as well. Some of the prediction equations, pixels used in these equations, number of modes these equations are used, number of pixels predicted by these equations and number of addition and shift operations performed by these prediction equations are shown in Table 2.1. Therefore, we propose calculating the common prediction equations for all 4x4 and 8x8 luminance angular prediction modes only once and using the results for the corresponding prediction modes.

There are 1792 prediction equations in 8x8 luminance angular prediction modes and 176 prediction equations in 4x4 luminance angular prediction modes. By using data reuse technique, the numbers of prediction equations that should be calculated for 8x8 and 4x4 luminance angular prediction modes are reduced to 560 and 71, respectively. As shown in Figure 2.3, an 8x8 PU and some of the 4x4 PUs have common neighboring pixels. They also have common prediction equations. Therefore, we used data reuse

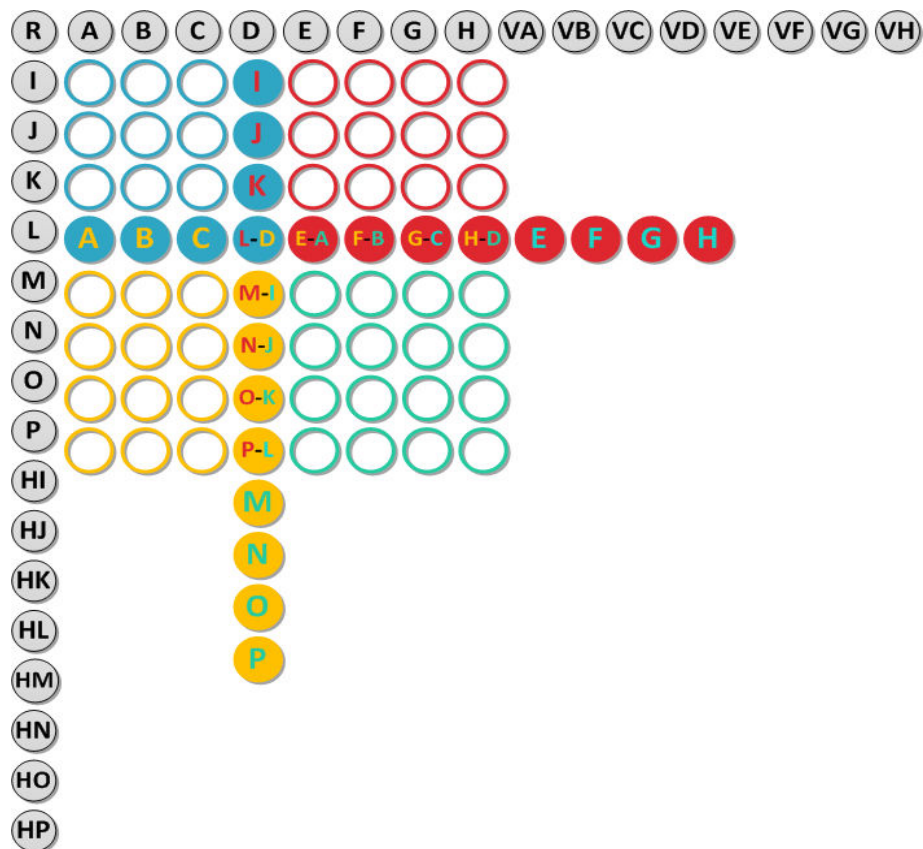


Figure 2.3 Neighboring Pixels of 4x4 and 8x8 PUs

technique for calculating predicted pixels of an 8x8 PU and predicted pixels of the corresponding four 4x4 PUs. In this way, the number of prediction equations that should be calculated for one 8x8 and four 4x4 PUs is reduced from 2496 to 721.

The computation reductions achieved by data reuse are shown in Table 2.2. 727372800 addition and 744194880 shift operations are performed by HEVC intra 4x4 and 8x8 luminance angular prediction modes for a full HD (1920x1080) frame. When data reuse technique is used, 115441920 addition and 117120960 shift operations are performed which corresponds to 84.12% and 84.26% reduction in addition and shift operations respectively.

We propose using PECR and PSCR techniques for HEVC intra prediction algorithm. PECR technique compares the pixels used in the prediction equations of angular intra prediction modes. If the pixels used in a prediction equation are equal, the predicted pixel by this equation is equal to these pixels. Therefore, this prediction equation simplifies to a constant value and prediction calculation for this equation

Table 2.1 Some of The HEVC Intra Prediction Equations

Pixels	Equations	PU Size	Used Modes	Pred. Pixels	# of Add.	# of Shift
I,J	$[27I+5J+16] \gg 5$	4x4	1	4	6	5
		8x8	3	9		
J,K	$[22J+10K+16] \gg 5$	4x4	2	5	5	6
		8x8	4	9		
K,L	$[17K+15L+16] \gg 5$	4x4	1	4	6	5
		8x8	1	6		
L,M	$[12L+20M+16] \gg 5$	4x4	3	7	4	5
		8x8	5	11		
M,N	$[6M+26N+16] \gg 5$	4x4	0	0	5	6
		8x8	4	6		
N,O	$[30N+2O+16] \gg 5$	4x4	0	0	5	6
		8x8	4	9		
O,P	$[8O+24P+16] \gg 5$	4x4	0	0	3	4
		8x8	5	12		
A,R	$[11A+21R+16] \gg 5$	4x4	1	2	6	5
		8x8	1	2		
A,B	$[5A+27B+16] \gg 5$	4x4	1	4	6	5
		8x8	1	6		
B,C	$[10B+22C+16] \gg 5$	4x4	2	5	5	6
		8x8	2	7		

Table 2.2 Computation Reductions by Data Reuse

Frame Size		4x4 Only		8x8 Only		One 8x8 and Four 4x4	
		# of Addition	# of Shift	# of Addition	# of Shift	# of Addition	# of Shift
1280 x 720	Original	50462720	50462720	121425920	128902400	323276800	330753280
	Data Reuse	21514240	20782080	39283200	40381440	51336356	52060566
	Reduction (%)	57.37%	58.59%	67.65%	68.67%	84.12%	84.26%
1920 x 1080	Original	113541120	11354112	273208320	290030400	727372800	744194880
	Data Reuse	48407040	46759680	88387200	90858240	115441920	117120960
	Reduction (%)	57.37%	58.59%	67.65%	68.67%	84.12%	84.26%

becomes unnecessary. PSCR technique also compares the pixels used in the prediction equations of angular intra prediction modes. If the pixels used in a prediction equation are similar, the predicted pixel by this equation is assumed to be equal to one of these pixels. PSCR technique determines the similarity of the pixels used in a prediction equation by truncating their least significant bits by the specified truncation amount (1, 2, 3, or 4 bits) and comparing the truncated pixels. If these truncated pixels are all equal, one of the original pixels is substituted in place of every pixel used in this prediction equation. Therefore, this prediction equation simplifies to a constant value and prediction calculation for this equation becomes unnecessary.

The number of prediction equations in intra luminance angular prediction modes with equal and similar pixels in a frame varies from frame to frame. We analyzed Tennis (1920x1080), Kimono (1920x1080), Vidyo1 (1280x720) and Vidyo3 (1280x720) videos [14] coded with quantization parameters (QP) 28, 35 and 42 using HEVC Test Model HM 5.2 encoder software [12], and determined how many prediction equations after using data reuse technique have equal pixels and similar pixels for different truncation amounts (1bT, 2bT, 3bT, 4bT) in one frame of each video sequence. The simulation results for some of the prediction equations for 8x8 PU size are shown in Tables 2.3 and 2.4.

We calculated the computation reductions achieved by PECC technique after data reuse and PSCR technique for 4bT after data reuse for one frame of each video sequence using the simulations results. As shown in Tables 2.5 and 2.6, PECC and PSCR techniques achieved up to 65% and 93% computation reductions, respectively. The proposed PECC and PSCR techniques have an overhead of only 2914560

comparisons for an HD (1280x720) frame and 6557760 comparisons for a full HD (1920x1080) frame.

We also quantified the impact of the proposed PSCR technique on the rate-distortion performance of the 4x4 intra prediction algorithm by using HEVC Test Model HM 5.2 encoder software. The rate distortion curves and average PSNR comparison of original 4x4 and 8x8 intra prediction algorithms and 4x4 and 8x8 intra prediction algorithms with the proposed PSCR technique for several HD and full HD size video frames are shown in Figure 2.4 and Table 2.7 respectively. The average PSNR values shown in Table 2.7 are calculated using the technique described in [15]. The proposed PSCR technique increases the PSNR slightly for some video frames and it decreases the PSNR slightly for some video frames.

Table 2.3 Percentages of 8x8 PUs with Equal and Similar Pixels for 1920x1080 Video Frames

		Equal (%)			Similar (1bT) (%)			Similar (2bT) (%)			Similar (3bT) (%)			Similar (4bT) (%)		
		QP			QP			QP			QP			QP		
Pixels		28	35	42	28	35	42	28	35	42	28	35	42	28	35	42
Tennis	L,J	45.6	42.9	46.7	62.0	60.0	62.3	75.7	75.4	76.8	84.8	84.0	86.2	91.0	90.7	91.6
	J,K	43.8	45.0	51.5	59.8	59.4	66.0	74.1	74.3	78.2	83.8	84.1	86.9	89.8	90.1	92.5
	K,L	44.9	45.8	52.7	61.0	61.1	66.6	74.8	75.2	78.6	84.9	85.2	87.0	90.5	90.8	92.5
	L,M	46.2	46.3	53.8	61.8	61.3	66.6	75.6	76.1	78.9	85.4	84.9	86.6	91.1	91.1	92.3
	A,R	62.9	68.6	72.8	74.5	77.9	80.4	84.7	85.9	87.2	91.3	91.8	91.9	94.9	95.1	95.0
	A,B	73.3	74.3	75.4	83.5	83.9	83.8	90.8	90.8	91.1	95.1	95.2	95.2	97.1	97.4	97.4
	B,C	77.5	79.6	81.0	85.8	87.1	87.9	92.0	92.6	93.1	95.8	96.1	96.5	97.7	97.8	98.1
	C,D	77.0	79.4	82.0	85.6	86.8	88.6	92.0	92.8	93.6	96.0	96.2	96.6	97.8	98.0	98.2
	D,E	77.2	79.3	81.8	85.7	87.0	88.2	92.0	92.6	93.2	95.7	96.3	96.5	97.8	98.0	98.2
	HI,HJ	79.4	82.2	83.0	71.0	70.9	73.0	81.2	81.2	83.5	88.2	88.2	90.2	93.0	92.9	93.8
VA,VB	58.4	58.2	62.9	87.3	88.7	89.0	93.1	93.6	93.9	96.5	96.7	96.7	98.0	98.1	98.2	
Kimono	L,J	42.3	41.5	45.1	52.2	53.8	57.1	64.1	66.9	70.7	76.0	78.7	81.9	85.9	87.6	89.8
	J,K	43.9	45.5	49.6	53.2	56.3	60.9	64.4	68.1	72.9	76.3	79.0	83.1	86.4	88.0	90.2
	K,L	43.9	46.0	51.1	53.8	57.0	61.8	65.3	68.8	73.5	77.1	79.7	83.3	86.3	88.4	91.0
	L,M	43.1	46.0	51.1	52.5	56.6	61.7	64.2	68.4	73.6	75.2	79.2	83.7	85.7	87.9	90.8
	A,R	39.9	42.0	47.8	49.6	52.9	57.6	61.0	64.4	68.7	73.9	75.8	78.6	84.3	85.1	86.7
	A,B	46.0	46.0	51.0	56.3	58.5	62.1	68.2	71.3	74.9	80.0	82.2	85.1	88.5	90.0	91.7
	B,C	47.7	50.0	55.4	57.4	61.1	66.4	68.6	72.7	77.5	80.0	82.8	86.9	88.7	90.3	92.8
	C,D	47.6	50.8	57.0	58.1	62.0	67.2	69.8	73.7	77.6	80.7	83.6	86.5	89.0	90.7	92.6
	D,E	46.9	50.6	57.2	56.6	61.6	67.2	68.3	72.8	77.7	79.7	83.1	86.4	88.2	90.5	92.4
	HI,HJ	56.9	57.5	61.0	64.4	66.3	69.5	73.4	75.5	79.0	82.2	84.4	87.0	89.5	91.0	92.7
VA,VB	56.4	57.5	62.4	64.6	67.3	70.9	74.2	77.4	80.6	83.7	85.9	88.4	90.1	92.0	93.5	

Table 2.4 Percentages of 8x8 PUs with Equal and Similar Pixels for 1280x720 Video Frames

		Equal (%)			Similar (1bT) (%)			Similar (2bT) (%)			Similar (3bT) (%)			Similar (4bT) (%)		
		QP			QP			QP			QP			QP		
Pixels		28	35	42	28	35	42	28	35	42	28	35	42	28	35	42
Vidyo 1	I,J	54.5	50.4	49.8	66.4	63.6	61.8	76.2	75.2	74.0	84.1	83.9	83.7	90.3	90.3	90.5
	J,K	57.4	56.9	47.1	67.9	67.7	66.9	77.1	77.2	76.7	84.7	84.8	84.9	90.2	90.6	91.1
	K,L	57.7	57.0	58.5	68.1	67.6	67.9	77.2	77.4	77.5	84.6	84.7	85.4	90.3	90.7	91.6
	L,M	57.5	57.3	58.7	67.2	67.8	67.9	76.2	77.2	77.4	84.0	84.6	84.8	89.9	90.5	90.6
	A,R	37.5	38.1	37.9	50.2	51.2	50.2	64.3	64.4	63.9	76.4	75.7	75.6	85.7	85.1	84.7
	A,B	43.8	41.1	38.6	57.4	56.6	53.4	70.0	70.6	69.3	80.9	81.5	81.3	88.6	89.2	89.1
	B,C	44.8	44.9	44.4	58.2	59.0	59.0	70.9	72.1	73.0	81.2	81.8	83.2	88.9	89.3	90.4
	C,D	45.7	45.3	46.3	59.0	59.8	60.4	72.1	73.1	73.9	81.8	82.9	83.5	89.8	90.1	90.6
	D,E	45.5	45.8	45.7	58.6	60.1	59.1	71.2	72.6	72.7	81.3	82.4	82.8	88.8	89.4	89.9
	HI,HJ	66.4	64.9	66.6	75.0	74.1	74.6	82.3	81.9	82.2	88.0	88.2	88.6	92.7	92.8	93.1
VA,VB	55.1	53.9	52.7	66.1	66.2	64.3	76.5	77.2	76.8	85.1	85.8	86.0	91.3	91.8	92.1	
Vidyo 3	I,J	60.5	59.8	62.2	68.8	69.0	70.4	76.1	75.6	78.7	82.9	82.6	85.4	89.1	89.2	91.0
	J,K	64.4	63.7	66.4	71.4	70.9	72.7	77.8	77.2	78.8	83.7	83.5	85.3	89.4	89.5	91.3
	K,L	63.9	63.9	66.8	70.8	71.3	73.6	77.2	78.1	80.0	83.5	84.3	86.3	89.6	90.1	91.7
	L,M	62.8	64.8	67.1	69.6	71.4	73.9	76.5	77.7	80.0	83.0	84.2	85.8	89.0	89.9	91.4
	A,R	51.3	53.0	51.1	59.9	61.0	59.2	68.7	68.4	67.8	76.7	75.8	75.9	83.1	82.8	82.4
	A,B	59.1	58.5	56.7	67.6	66.8	65.3	74.3	73.6	73.8	79.9	79.6	80.2	85.7	85.3	85.9
	B,C	62.7	61.9	61.1	69.4	68.6	67.9	75.3	75.0	76.0	80.6	81.1	81.6	86.2	86.8	87.2
	C,D	63.4	63.2	62.0	70.2	69.7	69.6	76.2	75.9	76.3	81.4	81.3	82.1	87.0	86.4	87.2
	D,E	62.7	61.5	62.1	69.2	69.2	69.3	75.2	75.2	75.8	81.0	81.1	81.6	86.6	86.8	87.4
	HI,HJ	71.0	71.6	73.7	77.1	77.9	79.1	82.6	82.7	85.0	87.3	87.4	89.5	92.0	92.1	93.5
VA,VB	67.6	68.3	66.8	74.2	74.4	73.3	79.6	79.6	80.0	83.9	84.1	84.9	88.7	88.6	89.4	

Table 2.5 Computation Reductions by PEQR After Data Reuse

QP		4x4 Only		8x8 Only		One 8x8 and Four 4x4	
		Addition Reduction	Shift Reduction	Addition Reduction	Shift Reduction	Addition Reduction	Shift Reduction
Vidyo1 (1280x720)	28	50.88%	50.87%	50.88%	50.87%	50.88%	52.99%
	42	50.04%	49.91%	51.02%	51.00%	50.82%	52.86%
Vidyo3 (1280x720)	28	61.82%	61.73%	61.82%	61.73%	61.84%	63.78%
	42	62.89%	62.81%	62.89%	62.81%	62.92%	64.91%
Tennis (1920x1080)	28	58.25%	58.08%	60.17%	60.09%	59.76%	59.66%
	42	62.92%	62.74%	65.73%	65.63%	65.11%	64.99%
Kimono (1920x1080)	28	44.75%	44.62%	46.15%	46.14%	45.84%	45.82%
	42	50.60%	50.47%	52.65%	52.62%	52.20%	52.16%

Table 2.6 Computation Reductions by PSCR After Data Reuse

QP		4x4 Only		8x8 Only		One 8x8 and Four 4x4	
		Addition Reduction	Shift Reduction	Addition Reduction	Shift Reduction	Addition Reduction	Shift Reduction
Vidyo1 (1280x720)	28	78.73%	78.33%	89.15%	88.94%	81.06%	80.94%
	42	80.81%	80.39%	89.73%	89.52%	82.40%	82.26%
Vidyo3 (1280x720)	28	78.41%	77.98%	87.40%	87.20%	80.56%	80.41%
	42	79.80%	79.36%	88.55%	88.34%	82.21%	82.04%
Tennis (1920x1080)	28	89.59%	89.06%	93.28%	93.05%	91.91%	91.58%
	42	90.43%	89.92%	93.55%	93.31%	92.31%	91.96%
Kimono (1920x1080)	28	77.21%	76.83%	86.34%	86.14%	83.76%	83.55%
	42	82.84%	82.42%	89.74%	89.54%	87.65%	87.39%

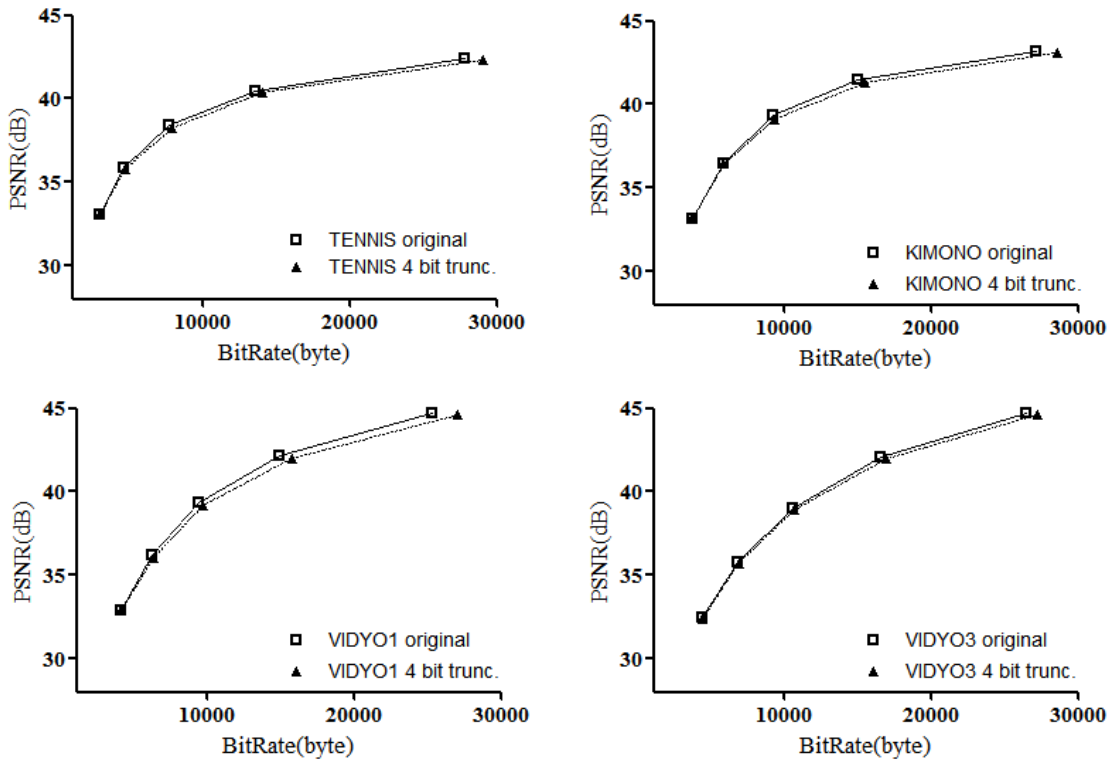


Figure 2.4 Rate Distortion Curves of Original 4x4 and 8x8 Intra Prediction Algorithms and 4x4 and 8x8 Intra Prediction Algorithms with PSCR Technique for 4bT

Table 2.7 Average PSNR Comparison of PSCR Technique

Frame	QP	Org. (dB)	1bT (dB)	Diff. (dB)	2bT (dB)	Diff. (dB)	3bT (dB)	Diff. (dB)	4bT (dB)	Diff. (dB)
Tennis (1920x1080)	28	40.108	40.105	-0.003	40.094	-0.014	40.053	-0.055	40.002	-0.106
	35	36.923	36.934	0.011	36.910	-0.013	36.890	-0.033	36.811	-0.112
	42	33.082	33.071	-0.011	33.111	0.029	33.073	-0.009	33.057	-0.025
Kimono (1920x1080)	28	41.063	41.069	0.006	41.042	-0.021	40.968	-0.095	40.901	-0.162
	35	37.666	37.638	-0.028	37.652	-0.014	37.603	-0.063	37.544	-0.122
	42	33.199	33.198	-0.001	33.234	0.035	33.196	-0.003	33.205	0.006
Vidyo1 (1280x720)	28	41.625	41.624	-0.001	41.608	-0.017	41.556	-0.069	41.482	-0.143
	35	37.411	37.412	0.001	37.409	-0.002	37.404	-0.007	37.336	-0.075
	42	32.911	32.902	-0.009	32.884	-0.027	32.887	-0.024	32.865	-0.046
Vidyo3 (1280x720)	28	41.480	41.493	0.013	41.458	-0.022	41.459	-0.021	41.389	-0.091
	35	37.127	37.117	-0.010	37.095	-0.032	37.105	-0.022	37.029	-0.098
	42	32.471	32.476	0.005	32.493	0.022	32.482	0.011	32.416	-0.055

2.3 Proposed HEVC Intra Prediction Hardware

The proposed HEVC intra prediction hardware implementing 16 angular prediction modes for 4x4 PU size and 33 angular prediction modes for 8x8 PU size including data reuse, PECR and PSCR techniques is shown in Figure 2.5.

Three local neighboring buffers are used to store neighboring pixels in the previously coded and reconstructed neighboring 4x4 and 8x8 luma PUs. After a luma PU in the current CU is coded and reconstructed, the neighboring pixels in this PU are stored in the corresponding buffers. These on chip neighboring buffers reduce the required off-chip memory bandwidth.

56 neighboring registers are used to store the neighboring pixels for the current one 8x8 and four 4x4 PUs. After these neighboring pixel registers are loaded in 16 cycles, five parallel datapaths are used to calculate the prediction equations for one 8x8 and four 4x4 PUs. The architecture of a datapath is shown in Figure 2.6. The predicted pixels are stored in the prediction equation register file.

The HEVC intra prediction hardware only including data reuse technique (IPHW+DR) does not have the comparison unit and the last multiplexer in the datapath. This hardware calculates the predicted pixels for one 8x8 and four 4x4 PUs in 160 clock cycles.

In the HEVC intra prediction hardware including both data reuse and PECR techniques (IPHW+DR+PECR), 56 8-bit comparators are used to check the equality of

the neighboring pixels. Based on the comparison results, disable signals are generated

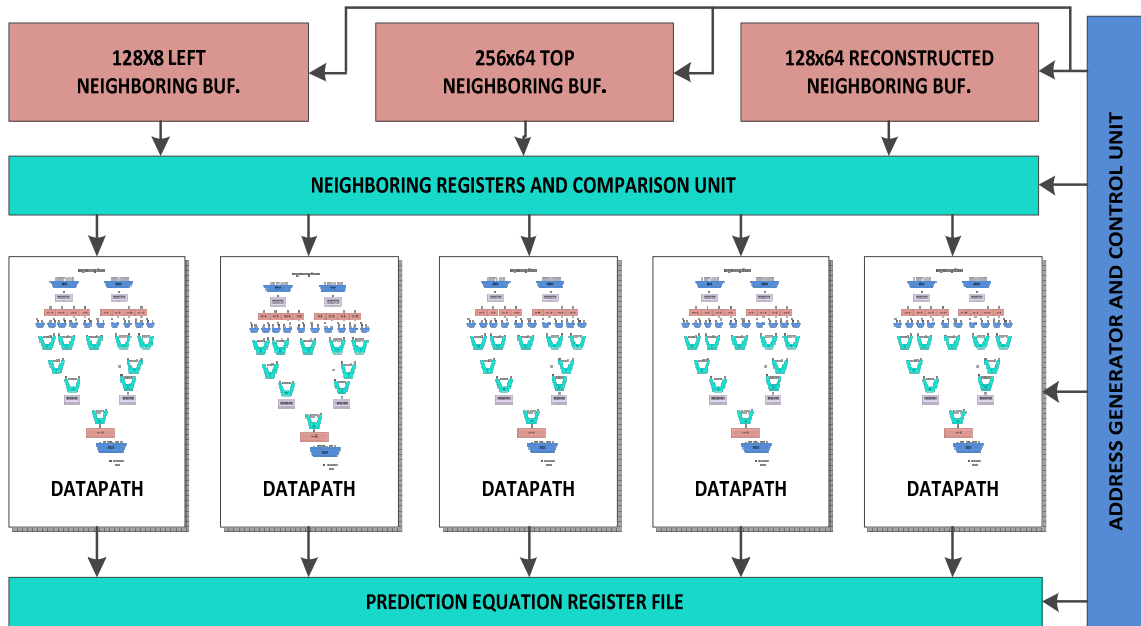


Figure 2.5 HEVC Intra Prediction Hardware

and sent to the datapaths implementing the prediction equations with equal pixels. If the neighboring pixels are equal, the last multiplexer in the datapath is used to select a neighboring pixel instead of the predicted pixel calculated by the datapath.

In the HEVC intra prediction hardware including both data reuse and PSCR techniques (IPHW+DR+PSCR), 56 comparators are used to check the similarity of the neighboring pixels. IPHW+DR+PSCR for 1bT uses 56 7-bit comparators. Similarly, IPHW+DR+PSCR for 4bT uses 56 4-bit comparators. Based on the comparison results, disable signals are generated and sent to the datapaths implementing the prediction equations with similar pixels. If the neighboring pixels are similar, the last multiplexer in the datapath is used to select a neighboring pixel instead of the predicted pixel calculated by the datapath.

IPHW+DR, IPHW+DR+PECR and IPHW+DR+PSCR are implemented using Verilog HDL. The hardware implementations are verified with RTL simulations using Mentor Graphics Modelsim SE. The RTL simulation results matched the results of a software model of HEVC intra prediction algorithm. The Verilog HDL codes are synthesized and mapped to a Xilinx XC6VLX75T FF784 FPGA with speed grade 3 using Xilinx ISE 12.3.

IPHW+DR FPGA implementation uses 1659 LUTs, 817 DFFs and 4 BRAMs.

IPHW+DR+PECR FPGA implementation uses 2381 LUTs, 849 DFFs, and 4 BRAMs. IPHW+DR+PSCR for 4bT FPGA implementation uses 2318 LUTs, 849 DFFs, and 4 BRAMs. All FPGA implementations are verified to work at 150 MHz by post place and route simulations. The FPGA implementation is also verified to work correctly on a Xilinx Virtex 6 FPGA board. Therefore, they can process 30 full HD (1920x1080) video frames per second.

IPHW+DR+PECR Verilog RTL code is also synthesized to Synopsys 90 nm standard cell library, and the resulting netlist is placed & routed. The resulting ASIC implementation works at 158 MHz, and its gate count is calculated as 5.4K according to NAND (2x1) gate area excluding on-chip memory.

We estimated the power consumptions of all FPGA implementations using Xilinx XPower tool for Tennis (1920x1080), Kimono (1920x1080), Vidyo1 (1280x720) and Vidyo3 (1280x720) videos [14]. In order to estimate the power consumption of a HEVC intra prediction hardware, timing simulation of its placed and routed netlist is done using Mentor Graphics ModelSim SE for one frame of each video sequence. The signal activities of these timing simulations are stored in VCD files, and these VCD files are used for estimating the power consumption of that HEVC intra prediction hardware using Xilinx XPower tool. Since HEVC intra prediction hardware is used as part of a HEVC video encoder, only internal power consumption is considered and input and output power consumptions are ignored. Therefore, the power consumption of a HEVC intra prediction hardware can be divided into four main categories; clock power, logic power, signal power and BRAM power.

The power and energy consumptions of IPHW+DR, IPHW+DR+PECR, and IPHW+DR+PSCR on this FPGA are shown in Table 2.8 and Table 2.9 for different QP values. As shown in these tables, PECR technique reduced the energy consumption of the proposed HEVC intra prediction hardware including data reuse technique up to 40%. PSCR technique reduced the energy consumption of the proposed HEVC intra prediction hardware including data reuse technique up to 66%.

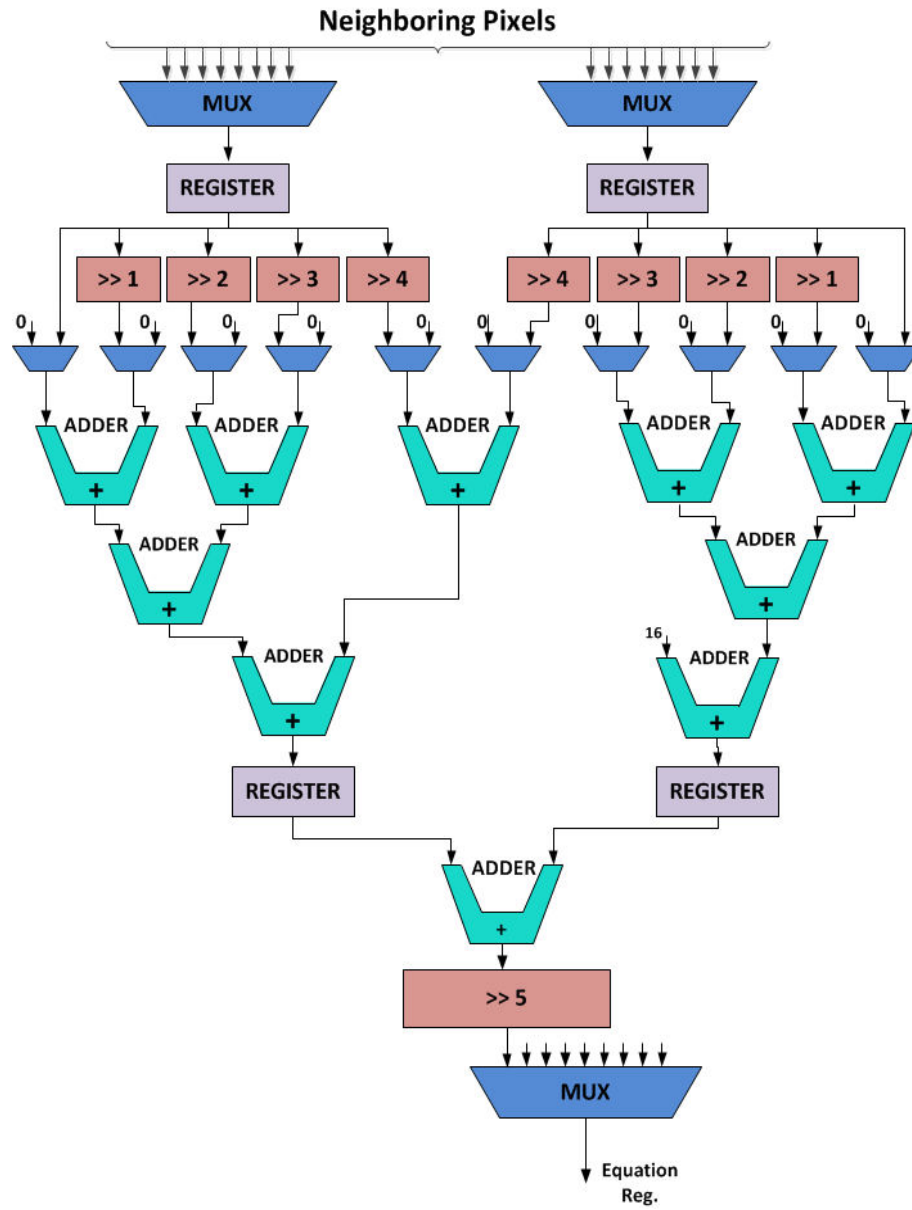


Figure 2.6 HEVC Intra Prediction Datapath

Table 2.8 Energy Consumption Reduction for 1920x1080 Video Frames

Frames	Category	Intra Pred. Hardware		Intra Pred. Hardware with PECR		Intra Pred. Hardware with PSCR (1bT)		Intra Pred. Hardware with PSCR (2bT)		Intra Pred. Hardware with PSCR (3bT)		Intra Pred. Hardware with PSCR (4bT)	
		QP 28	QP 42	QP 28	QP 42	QP 28	QP 42	QP 28	QP 42	QP 28	QP 42	QP 28	QP 42
Tennis	Time (ms)	42.101	42.101	32.180	31.467	28.322	24.338	23.814	22.388	20.532	20.101	18.588	18.391
	Clock	13.27	13.27	17.12	16.35	14.81	14.67	14.21	14.13	14.10	14.06	13.91	13.89
	Logic	13.87	13.43	9.37	8.99	8.60	8.16	7.98	7.86	7.91	7.69	7.73	7.43
	Signal	14.48	14.18	8.48	7.94	9.01	8.51	8.98	8.33	8.13	7.69	8.03	7.49
	BRAM	2.98	2.87	2.98	3.17	3.38	3.97	3.38	3.56	3.99	4.17	4.49	4.57
	Power	44.6	43.75	37.95	36.45	35.80	35.31	34.55	33.88	34.13	33.65	34.16	33.38
	Energy	1877.7	1841.9	1221.2	1146.9	1013.9	859.4	822.8	758.5	700.8	676.4	634.9	613.9
	Reduction			34.96%	37.73%	46.00%	53.34%	56.18%	58.82%	62.67%	63.29%	66.19%	66.67%
Kimono	Time (ms)	42.101	42.101	33.427	31.890	31.681	24.338	28.391	25.996	24.657	22.563	21.113	19.794
	Clock	13.27	13.27	17.17	16.84	14.89	14.67	14.85	14.58	14.29	14.19	14.04	13.98
	Logic	13.78	13.89	10.14	9.33	9.97	8.26	9.00	8.21	8.74	8.10	8.22	7.70
	Signal	14.34	14.01	9.27	8.58	10.61	9.51	10.42	9.18	8.77	8.08	8.41	7.81
	BRAM	2.98	2.87	2.97	2.97	3.18	3.97	3.18	3.38	3.48	3.57	4.29	4.41
	Power	44.37	44.04	39.55	37.72	38.65	36.41	37.45	35.35	35.28	33.94	34.96	33.90
	Energy	1868.0	1854.1	1322.0	1202.8	1224.5	886.1	1063.2	918.9	869.9	765.8	738.1	671.0
	Reduction			29.23%	35.12%	34.45%	52.21%	43.08%	50.44%	53.43%	58.70%	60.49%	63.81%

Table 2.9 Energy Consumption Reduction for 1280x720 Video Frames

Frames	Category	Intra Pred. Hardware		Intra Pred. Hardware with PECR		Intra Pred. Hardware with PSCR (1bT)		Intra Pred. Hardware with PSCR (2bT)		Intra Pred. Hardware with PSCR (3bT)		Intra Pred. Hardware with PSCR (4bT)	
		QP 28	QP 42	QP 28	QP 42	QP 28	QP 42	QP 28	QP 42	QP 28	QP 42	QP 28	QP 42
Vidyo 1	Time (ms)	18.711	18.711	15.134	13.425	13.498	12.893	11.793	11.702	10.256	10.217	9.012	8.986
	Clock	13.27	13.27	15.31	15.27	14.36	14.34	14.03	13.94	13.58	13.54	13.38	13.38
	Logic	12.68	12.39	10.14	9.45	9.80	9.41	9.00	8.75	8.51	8.32	8.22	7.98
	Signal	13.95	13.79	9.77	9.06	9.74	9.28	8.81	8.47	8.06	7.79	7.95	7.64
	BRAM	2.77	2.87	2.77	3.17	3.17	3.37	3.77	3.77	4.38	4.37	4.79	4.77
	Power	42.67	42.32	37.99	36.95	37.07	36.4	35.61	34.93	34.53	34.02	34.34	33.77
	Energy	798.4	791.8	574.9	496.1	500.4	469.3	419.8	408.8	354.1	347.6	309.5	303.5
	Reduction			27.99%	37.35%	37.32%	40.73%	47.41%	48.37%	55.65%	56.1%	61.3%	61.67%
Vidyo 3	Time (ms)	18.711	18.711	13.843	12.766	12.615	12.137	11.541	11.301	10.519	10.278	9.488	9.264
	Clock	13.28	13.28	15.96	15.78	15.27	15.21	14.34	14.31	13.60	13.56	13.44	13.39
	Logic	12.44	12.26	9.46	9.19	9.28	8.83	8.61	8.31	8.27	8.02	8.02	7.80
	Signal	14.08	13.86	9.36	8.98	9.31	8.75	8.44	8.03	7.81	7.47	7.69	7.40
	BRAM	2.87	3.07	3.17	3.38	3.37	3.57	3.77	3.97	4.17	4.37	4.48	4.67
	Power	42.69	42.47	37.95	37.33	37.23	36.36	35.16	34.62	33.85	33.42	33.63	33.26
	Energy	798.8	794.7	525.3	476.6	469.7	441.3	405.8	391.2	356.1	343.5	319.1	308.1
	Reduction			34.3%	40.02%	40.33%	44.47%	49.2%	50.77%	55.42%	56.78%	60.1%	61.23%

2.4 HEVC Intra Prediction Hardware Implementation on FPGA Board

In this thesis, the IPHW+DR+PECR and IPHW+DR+PSCR hardwares are implemented on a ML605 FPGA board which includes a Virtex 6 XC6VLX240T FPGA, 512 MB DDR RAM and 32 MB Flash memory and interfaces such as UART and DVI.

A software running on MicroBlaze processor is developed to transfer the inputs of the intra prediction hardware from a host computer in an appropriate order and to gather the outputs of the hardware for sending them back to the host computer and displaying the resulting frame on a monitor. The intra prediction hardware is added as a peripheral to a bus where the MicroBlaze processor is the master. For this purpose the intra prediction hardware is modified to be a slave peripheral for this data bus and 8 software accessible registers are added to the hardware. 2 of these registers are used by the software running on MicroBlaze for writing the inputs to the hardware and the other 2 are used for gathering the outputs and the status information from the hardware.

The software gets one input frame from the host computer using the UART interface and writes it to a DDR RAM. Then, it loads the BRAMs of the intra prediction hardware with the reference pixels. After the intra prediction hardware generates the done signal, the software reads the intra-coded pixels updated by the intra prediction hardware and writes them to the DDR RAM. This process is repeated for all the CUs. Finally, the intra coded frame is displayed on a monitor using the DVI interface as shown in Figure 2.7. The top figure shows the output of intra prediction hardware, the middle one shows the original frame, and the bottom one shows the output of HEVC HM encoder software.

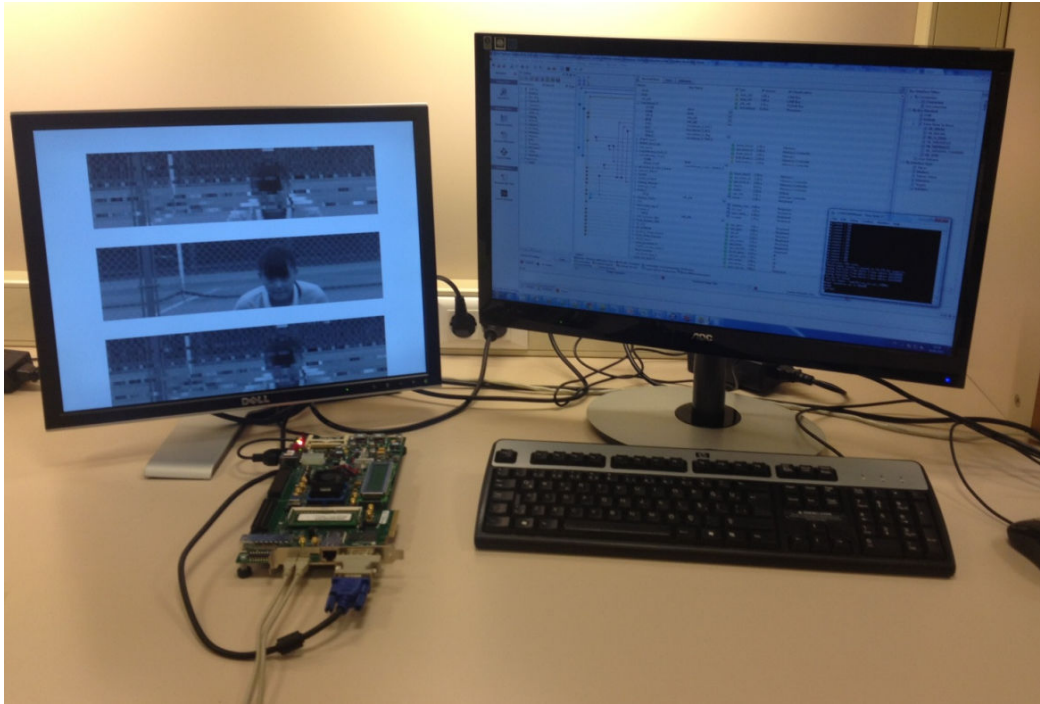


Figure 2.7 HEVC Intra Prediction Hardware FPGA Board Implementation

CHAPTER III

A HIGH PERFORMANCE AND LOW ENERGY INTRA PREDICTION HARDWARE FOR HEVC VIDEO DECODER

Joint collaborative team on video coding (JCT-VC) recently developed a new video compression standard called High Efficiency Video Coding (HEVC) [1]. HEVC provides 37% better coding efficiency than H.264 which is the current state-of-the-art video compression standard. HEVC also provides 23% bit rate reduction for the intra prediction only case [4, 17].

Intra prediction algorithm predicts the pixels of a block from the pixels of its already coded and reconstructed neighboring blocks. In H.264 standard, there are 9 intra prediction modes for 4x4 luminance blocks, and 4 intra prediction modes for 16x16 luminance blocks [18]. In HEVC, for the luminance component of a frame, intra prediction unit (PU) sizes can be from 4x4 up to 64x64 and number of intra prediction modes for a PU can be up to 35 [1, 19].

Since the intra prediction algorithm in HEVC standard requires significantly more computations than the intra prediction algorithm in H.264 standard, in this thesis, we propose novel techniques for reducing amount of computations performed by intra prediction algorithm in HEVC decoder without any PSNR and bit rate loss, and therefore reducing energy consumption of intra prediction hardware in HEVC decoder.

Data reuse techniques are proposed for reducing amount of computations performed by H.264 intra prediction algorithm in [20, 21]. In this thesis, we propose using data reuse technique for intra prediction algorithm in HEVC decoder. In HEVC,

intra 4x4 and 8x8 luminance prediction modes have identical equations. Therefore, we propose calculating the common prediction equations for all 4x4 and 8x8 luminance prediction modes only once and using the results for the corresponding prediction modes. The simulation results obtained by HEVC Test Model HM 5.2 decoder software [12] for several benchmark videos showed that this technique achieved more than 60% computation reduction.

Pixel equality and similarity based techniques are proposed for reducing amount of computations performed by H.264 intra prediction algorithm in [9, 10, 22]. In this thesis, we propose using pixel equality based computation reduction (PECR) technique for intra prediction algorithm in HEVC decoder. PECR technique compares the pixels used in the prediction equations of intra prediction modes. If the pixels used in a prediction equation are equal, the predicted pixel by this equation is equal to these pixels. Therefore, this prediction equation simplifies to a constant value and prediction calculation for this equation becomes unnecessary. The simulation results obtained by HEVC Test Model HM 5.2 decoder software [12] for several benchmark videos showed that using this technique after data reuse achieved more than 40% computation reduction with a small comparison overhead.

We also designed a high performance intra prediction hardware for angular prediction modes of 4x4 and 8x8 PU sizes including the proposed techniques for HEVC video decoding. The proposed hardware is implemented using Verilog HDL. The Verilog RTL code is mapped to a Xilinx Virtex 6 FPGA, and it is verified to work at 166.7 MHz by post place & route simulations. The proposed FPGA implementation, in the worst case, can process 100 full HD (1920x1080) video frames per second. We quantified the impact of PECR technique on the energy consumption of the proposed intra prediction hardware for HEVC video decoding including data reuse technique on this FPGA using Xilinx XPower Analyzer tool, and PECR technique reduced its energy consumption more than 40% [16].

An intra prediction hardware for HEVC video decoding is not reported in the literature. An intra prediction hardware only for 4x4 PU size for HEVC video encoding is presented in [13]. However, no power reduction technique is used in this hardware, and its power consumption is not reported. A parallel HEVC decoder software is presented in [23].

3.1 Proposed Computation Reduction Techniques

In HEVC, intra 4x4 and 8x8 luminance prediction modes have identical equations. Some of the prediction equations, pixels used in these equations, number of modes these equations are used, number of pixels predicted by these equations and number of addition and shift operations performed by these prediction equations are shown in Table 3.1.

Since intra prediction algorithm used in HEVC decoder has to find the intra prediction only for the prediction mode selected by HEVC encoder, in this thesis, we propose calculating the common prediction equations for each 4x4 and 8x8 luminance prediction mode only once and using the results for the corresponding prediction mode.

Each angular 8x8 intra prediction mode has 64 prediction equations, except 5 angular 8x8 intra prediction modes which have no prediction equations. Each angular 4x4 intra prediction mode has 16 prediction equations, except 5 angular 4x4 intra prediction modes which have no prediction equations. When data reuse technique is used, at least 8 prediction equations and at most 56 prediction equations are calculated for angular 8x8 intra prediction modes instead of calculating 64 prediction equations for each mode. Similarly, when data reuse technique is used, at least 4 prediction equations and at most 12 prediction equations are calculated for angular 4x4 intra prediction modes instead of calculating 16 prediction equations for each mode. 8 prediction equations are calculated for modes 22, 23, 30 and 31 of 8x8 PU size, and 4 prediction equations are calculated for modes 12, 13, 16 and 17 of 4x4 PU size.

We decoded Tennis (1920x1080), Basketball Drive (1920x1080), Vidyol (1280x720) and Vidyol3 (1280x720) videos [14] coded with quantization parameter (QP) 22, 27 and 32 using HEVC Test Model HM 5.2 decoder software [12], and determined the PU sizes and intra prediction modes selected by HEVC Test Model HM 5.2 encoder software [12] which is modified to use only 4x4 and 8x8 prediction modes for intra prediction. The results for one frame from each video sequence are shown in Figure 3.1 and Figure 3.2, respectively. Since 8x8 PU size is selected more often than 4x4 PU size, data reuse technique achieves more computation reductions for 8x8 PU size.

Table 3.1 Some of The HEVC Intra Prediction Equations

Pixels	Equations	PU Size	Pred. Pixels	# of Add.	# of Shift
I,J	$[5I + 27J + 16] \gg 5$	4x4	4	6	5
		8x8	6		
J,K	$[10J + 22K + 16] \gg 5$	4x4	4	5	6
		8x8	6		
K,L	$[15K + 17L + 16] \gg 5$	4x4	4	6	5
		8x8	6		
L,M	$[20L + 12M + 16] \gg 5$	4x4	4	4	5
		8x8	6		
M,N	$[25M + 7N + 16] \gg 5$	4x4	-	6	5
		8x8	6		
R,I	$[5R + 27I + 16] \gg 5$	4x4	-	6	5
		8x8	2		
N,O	$[3N + 29O + 16] \gg 5$	4x4	-	6	5
		8x8	2		
O,P	$[3O + 29P + 16] \gg 5$	4x4	-	6	5
		8x8	6		
L,M	$[25L + 7M + 16] \gg 5$	4x4	-	6	5
		8x8	2		
I,J	$[10I + 22J + 16] \gg 5$	4x4	-	5	6
		8x8	2		

The computation reductions achieved by data reuse technique for these video frames (one frame from each video sequence) are shown in Table 3.2 and 3.3. For Tennis (1920x1080) video frame coded with QP 27, 6162968 addition and 6834328 shift operations are performed by intra luminance prediction modes for HEVC video decoding. For the same video frame with same QP, when data reuse technique is used, 1744358 addition and 1848102 shift operations are performed by intra luminance prediction modes for HEVC video decoding. This corresponds to 71.70% and 72.96% reduction in addition and shift operations, respectively.

In this thesis, we propose using PECR technique for intra prediction algorithm in HEVC decoder. PECR technique compares the pixels used in the prediction equations of intra prediction modes. If the pixels used in a prediction equation are equal, the predicted pixel by this equation is equal to these pixels. Therefore, this prediction equation simplifies to a constant value and prediction calculation for this equation becomes unnecessary.

The number of intra prediction equations with equal pixels in a frame varies from frame to frame. We decoded Tennis (1920x1080), Basketball Drive (1920x1080), Vidyo1 (1280x720) and Vidyo3 (1280x720) videos [14] coded with QP 22, 27 and 32

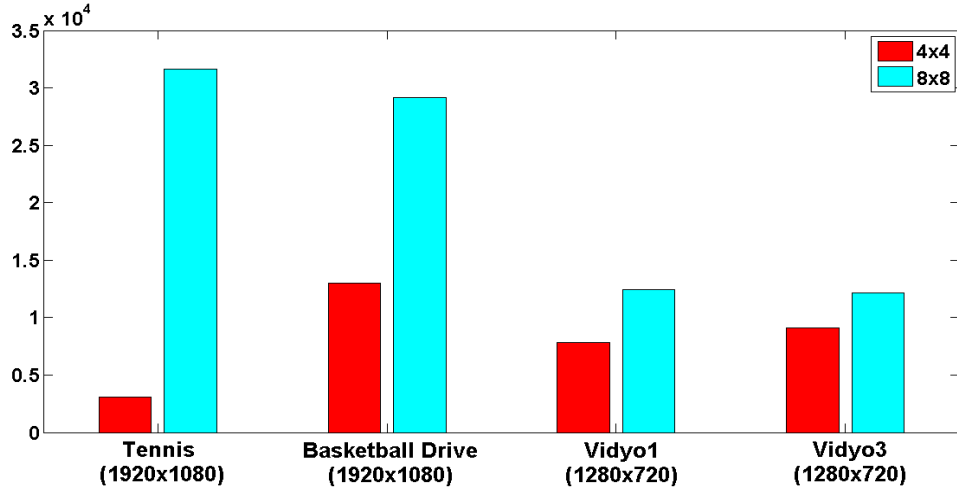


Figure 3.1 PU Sizes Selected by HEVC Video Encoder for Intra Prediction (QP = 27)

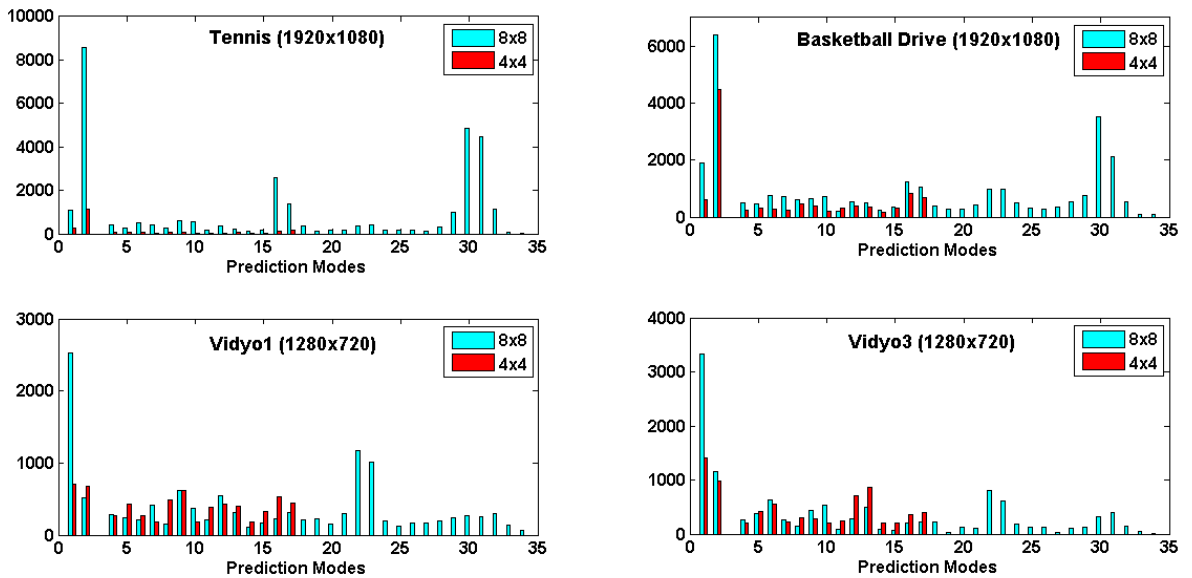


Figure 3.2 Prediction Modes Selected by HEVC Video Encoder for Intra Prediction (QP = 27)

using HEVC Test Model HM 5.2 decoder software [12], and determined how many prediction equations after using data reuse technique have equal pixels in one frame of each video sequence coded by HEVC Test Model HM 5.2 encoder software [12] which is modified to use only 4x4 and 8x8 prediction modes for intra prediction. The simulation results for some of the prediction equations for 8x8 PU size are shown in Table 3.4.

We calculated the computation reductions achieved by PECR technique after data reuse for one frame of each video sequence using the simulations results. As shown in Table 3.5, PECR technique after data reuse achieved more than 21.80% computation reduction.

Table 3.2 Computation Reductions by Data Reuse for 1920 x 1080 Frames

Size		1920 x 1280					
Frame		Tennis			Basketball Drive		
QP		22	27	32	22	27	32
# of Add.	Original	6008264	6162968	6156076	6347644	6029036	6379668
	Data Reuse	1734011	1744358	2008967	2241783	2117649	2147178
	Reduction	71.14 %	71.70 %	67.37 %	64.68 %	64.88 %	66.34 %
# of Shift	Original	6629768	6834328	6718380	6887228	6575404	6936340
	Data Reuse	1834547	1848102	2098215	2348303	2232217	2255018
	Reduction	72.33 %	72.96 %	68.77 %	65.90 %	66.05 %	67.49 %

Table 3.3 Computation Reductions by Data Reuse for 1280 x 720 Frames

Size		1280 x 720					
Frame		Vidyo 1			Vidyo 3		
QP		22	27	32	22	27	32
# of Add.	Original	3009144	2881768	2778860	2305208	2370884	2505556
	Data Reuse	1195890	1144221	1102014	912230	897531	937865
	Reduction	60.26 %	60.29 %	60.34 %	60.43 %	62.14 %	62.57 %
# of Shift	Original	3209464	3095336	2968748	2434104	2521284	2671252
	Data Reuse	1244690	1201397	1149990	944422	926507	966017
	Reduction	61.22 %	61.19 %	61.26 %	61.20 %	63.25 %	63.84 %

Table 3.4 Percentages of 8x8 PUs with Equal Pixels

Pixels	Tennis (%)		Basketball Drive (%)		Vidyo1 (%)		Vidyo3 (%)	
	QP 22	QP 32	QP 22	QP 32	QP 22	QP 32	QP 22	QP 32
I,J	42.9	41.9	18.5	36.9	55.6	48.6	60.7	59.3
J,K	42.1	57.3	15.4	45.6	36.3	61.8	62.8	67.1
K,L	41.2	58.6	18.3	46.2	57.2	62.3	61.6	68.0
L,M	42.8	61.0	19.1	47.9	56.6	63.3	62.3	68.4
M,N	41.3	60.4	18.7	47.4	55.8	63.6	61.5	69.4
A,R	50.0	58.9	24.3	40.1	28.5	32.1	59.2	44.8
A,B	71.6	64.6	33.1	51.4	42.9	38.0	59.2	52.3
B,C	78.5	75.0	35.9	65.3	43.8	46.6	61.6	59.9
C,D	76.7	81.7	34.3	66.8	42.7	51.4	60.6	63.3
D,E	76.8	81.9	34.9	67.5	43.5	52.3	60.7	63.4
HI,HJ	58.8	54.3	38.8	52.6	66.4	61.7	71.0	69.7
VA,VB	56.9	71.7	45.9	61.1	54.4	50.6	67.6	62.1

Table 3.5 Computation Reductions (%) by PECR After Data Reuse

Size	1920x1080						1280x720					
Frame	Tennis			Basketball Drive			Vidyo 1			Vidyo 3		
QP	22	27	32	22	27	32	22	27	32	22	27	32
Addition	46.81	48.82	54.31	26.50	40.39	40.45	42.97	39.64	38.76	56.25	55.65	56.04
Reduction												
Shift	45.14	48.09	53.56	21.80	38.63	39.65	38.49	37.10	36.93	50.04	51.06	50.29
Reduction												

The proposed PECR technique has to perform at least 8 and at most 12 comparisons for 8x8 intra prediction modes and at least 4 and at most 5 comparisons for 4x4 intra prediction modes. Table 3.6 shows the number of comparisons performed by PECR technique, the number of addition reductions achieved by PECR technique, and the percentage of the comparisons to the addition reductions. As shown in the table, the overhead of comparing the pixels used in the prediction equations is much smaller than the amount of addition reductions achieved by PECR technique.

Table 3.6 Comparison Overhead

Frame	QP	# of Comparison	Addition Reduction	%
Tennis	22	173214	1734041	9.98
	32	175360	2008967	8.73
Basketball Drive	22	205297	2241783	9.16
	32	184855	2147178	8.61
Vidyo1	22	98365	1195890	8.23
	32	84919	1102014	7.71
Vidyo3	22	79286	912230	8.69
	32	76315	937865	8.14

3.2 Proposed Intra Prediction Hardware Architecture and Its Energy Consumption

The proposed intra prediction hardware for HEVC video decoding implementing 16 angular prediction modes for 4x4 PU size and 33 angular prediction modes for 8x8 PU size including data reuse and PECR techniques is shown in Figure 3.3. The proposed intra prediction hardware generates the predicted pixels for the luma component of a PU using the luma prediction mode selected by HEVC encoder.

Three local neighboring buffers are used to store neighboring pixels in the previously coded and reconstructed neighboring 4x4 and 8x8 luma PUs. After a luma PU in the current CU is coded and reconstructed, the neighboring pixels in this PU are stored in the corresponding buffers. These on-chip neighboring buffers reduce the required off-chip memory bandwidth.

56 neighboring registers are used to store the neighboring pixels for the current 8x8 and 4x4 PUs. After these neighboring pixel registers are loaded in 16 cycles, 15x8 reference main array is loaded with the necessary neighboring pixels for the given prediction mode. Two parallel datapaths are used to calculate the prediction equations for 8x8 and 4x4 PUs. The architecture of a datapath is shown in Figure 3.4. The decoded pixels are stored in the prediction equation register file.

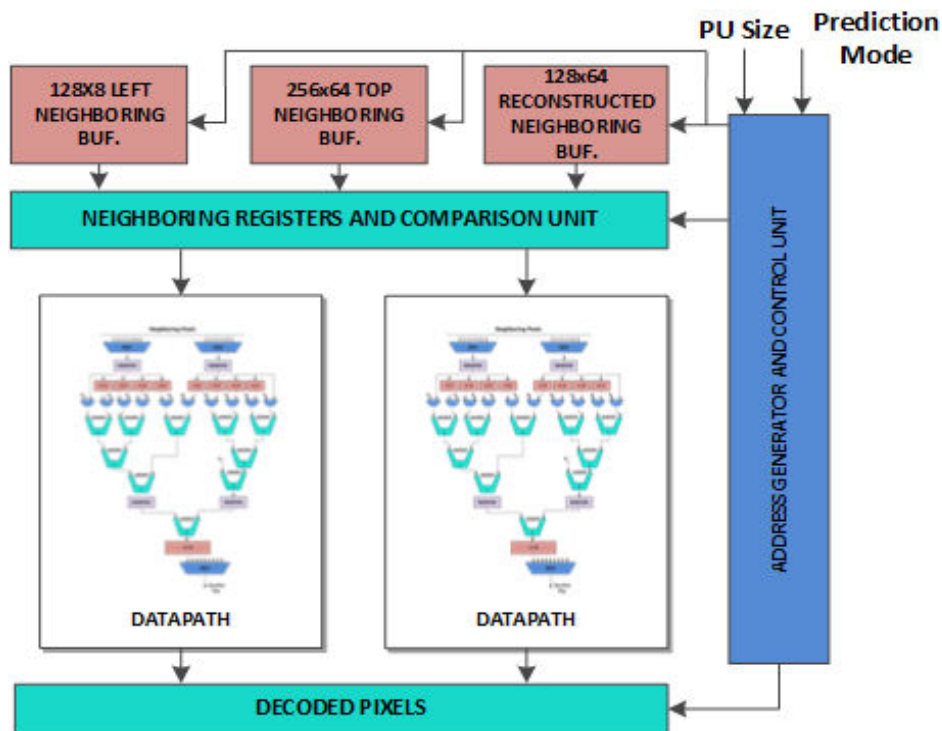


Figure 3.3 Intra Prediction Hardware for HEVC Video Decoding

The intra prediction hardware (IPHW) does not have the comparison unit and the last multiplexer in the datapath. This hardware calculates the predicted pixels for 8x8 and 4x4 PUs in 48 and 12 clock cycles respectively. In the intra prediction hardware including both data reuse and PECR techniques (IPHW+DR+PECR), 8-bit comparators are used to check the equality of the neighboring pixels. Based on the

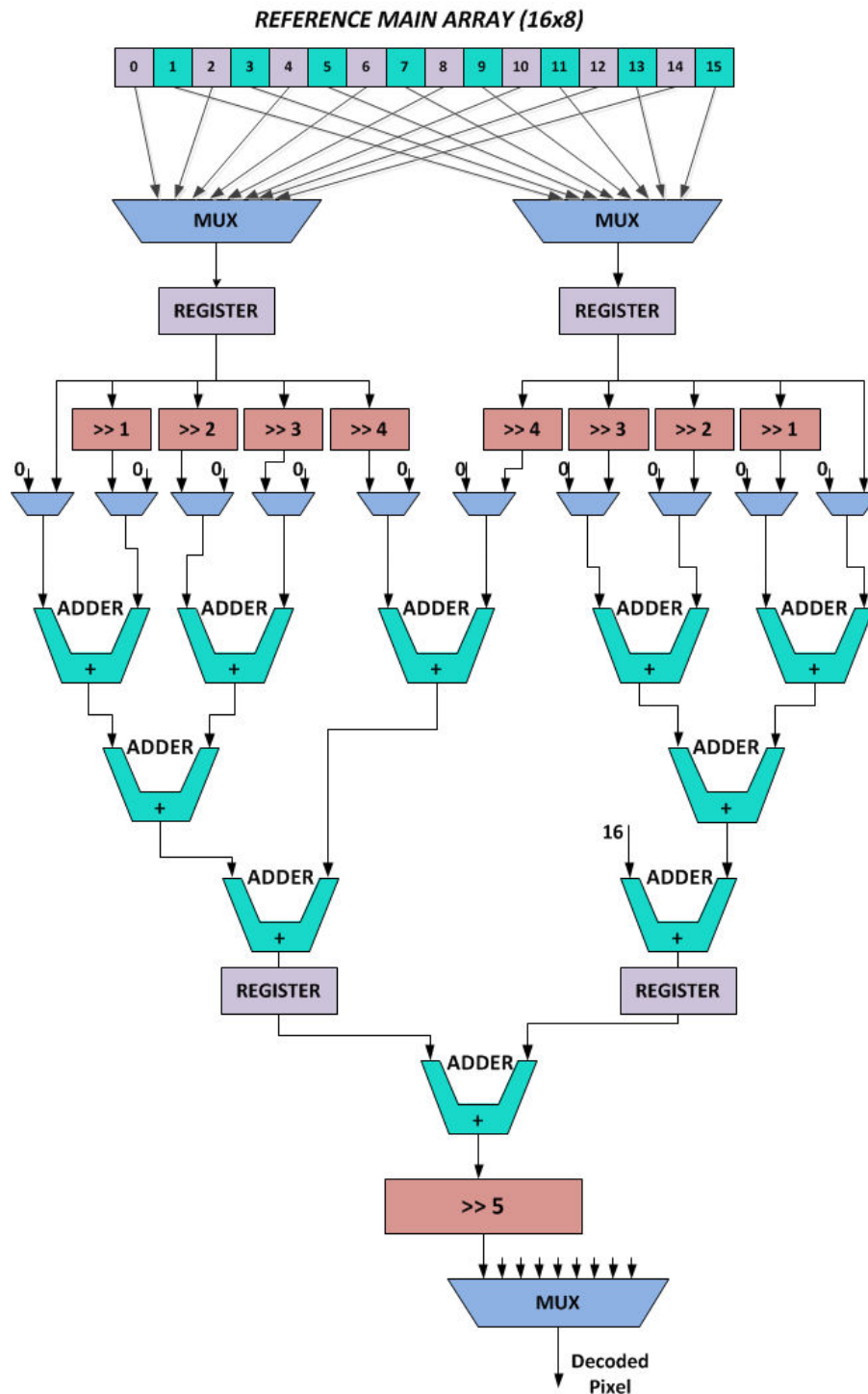


Figure 3.4 Intra Prediction Datapath

comparison results, disable signals are generated and sent to the datapaths implementing the prediction equations with equal pixels. If the neighboring pixels are equal, the last multiplexer in the datapath is used to select a neighboring pixel instead of the predicted pixel calculated by the datapath.

Both IPHW and IPHW+DR+PECR are implemented using Verilog HDL. The hardware implementations are verified with RTL simulations using Mentor Graphics Modelsim SE. The RTL simulation results matched the results of a MATLAB model of intra prediction algorithm in HEVC decoder.

The Verilog HDL codes are synthesized and mapped to a Xilinx XC6VLX75T FF784 FPGA with speed grade 3 using Xilinx ISE 12.3. IPHW FPGA implementation uses 3482 LUTs, 740 DFFs and 4 BRAMs. IPHW+DR+PECR FPGA implementation uses 4835 LUTs, 746 DFFs, and 4 BRAMs. Both FPGA implementations are verified to work at 166.7 MHz by post place & route simulations. IPHW FPGA implementation, in the worst-case, can process 90 full HD (1920x1080) video frames per second, and IPHW+DR+PECR FPGA implementation, in the worst-case, can process 100 full HD (1920x1080) video frames per second.

We estimated the power consumptions of both FPGA implementations using Xilinx XPower Analyzer tool for Tennis (1920x1080), Basketball Drive (1920x1080), Vidy01 (1280x720) and Vidy03 (1280x720) videos [14] coded with QP 22, 27 and 32. In order to estimate the power consumption of an intra prediction hardware, timing simulation of its placed and routed netlist is done at 166.7 MHz using Mentor Graphics ModelSim SE for one frame of each video sequence. The signal activities of these timing simulations are stored in VCD files, and these VCD files are used for estimating the power consumption of the intra prediction hardware for HEVC video decoding using Xilinx XPower Analyzer tool. Since the proposed intra prediction hardware will be used as part of a HEVC decoder, only internal power consumption is considered and input and output power consumptions are ignored.

The energy consumptions of the intra prediction hardware for HEVC video decoding and the intra prediction hardware for HEVC video decoding including both data reuse and PECR techniques on this FPGA are shown in Table 3.7 and Table 3.8 for different size video frames. As shown in these tables, data reuse and PECR techniques reduced the energy consumption of the proposed intra prediction hardware for HEVC video decoding up to 42.78%.

Table 3.7 Energy Consumption Reduction for 1280x720 Video Frames

QP	Vidyo 1						Vidyo 3					
	22		27		32		22		27		32	
	Org.	Low Energy	Org.	Low Energy	Org.	Low Energy	Org.	Low Energy	Org.	Low Energy	Org.	Low Energy
Clock(mW)	19.70	18.77	19.71	18.80	19.71	18.81	19.59	18.70	19.63	18.71	19.66	18.72
Logic(mW)	8.89	7.90	8.96	7.98	8.86	7.95	7.83	6.46	7.98	6.56	8.20	6.75
Signal(mW)	14.11	14.82	14.12	14.86	13.98	14.74	13.67	12.79	13.95	12.80	14.24	12.93
BRAM(mW)	23.50	19.45	23.24	19.18	23.23	19.16	24.14	20.55	24.58	20.26	24.04	20.24
Total Time(ms)	4.294	3.522	4.054	3.187	3.826	2.972	3.819	3.358	3.765	3.133	3.708	2.860
Total Power(mW)	66.2	60.94	66.03	60.82	65.78	60.66	65.23	58.5	66.14	58.33	66.14	58.64
Energy (uJ)	284.3	214.6	267.7	193.8	251.7	180.3	249.1	196.5	249.0	182.8	245.3	167.7
Energy Red.	24.50 %		27.59 %		28.37 %		21.14 %		26.61 %		31.62 %	

Table 3.8 Energy Consumption Reduction for 1920x1080 Video Frames

QP	Tennis						Basketball Drive					
	22		27		32		22		27		32	
	Org.	Low Energy	Org.	Low Energy	Org.	Low Energy	Org.	Low Energy	Org.	Low Energy	Org.	Low Energy
Clock(mW)	21.17	21.03	21.08	21.31	21.07	21.15	21.03	21.04	21.16	21.04	21.08	21.06
Logic(mW)	9.64	9.27	9.33	7.37	9.41	7.04	9.46	7.42	9.67	7.72	9.68	7.58
Signal(mW)	16.71	15.64	15.49	14.21	15.63	13.39	16.57	14.44	15.95	14.49	15.99	14.66
BRAM(mW)	23.21	20.06	21.44	20.67	23.13	20.43	23.58	19.79	23.29	19.80	22.46	19.50
Total Time(ms)	8.93	5.71	8.36	5.36	8.28	5.29	11.18	7.86	9.65	6.54	9.00	5.89
Total Power(mW)	70.73	66.00	67.34	63.56	69.24	62.01	70.64	62.69	70.07	63.05	69.21	62.8
Energy (uJ)	631.3	376.8	563.0	340.8	573.0	327.9	789.7	492.5	676.3	412.5	622.9	369.8
Energy Red.	40.31%		39.47%		42.78%		37.63%		39.00%		40.63%	

CHAPTER IV

A RECONFIGURABLE HEVC SUB-PIXEL INTERPOLATION HARDWARE

In order to increase the performance of integer pixel motion estimation, sub-pixel (half and quarter) accurate variable block size motion estimation is performed in HEVC. Sub-pixel interpolation is one of the most computationally intensive parts of HEVC video encoder and decoder. In the high efficiency and low complexity configurations of HEVC decoder, 37% and 50% of the HEVC decoder complexity is caused by sub-pixel interpolation on average, respectively [24].

In H.264 standard, a 6-tap FIR filter is used for half-pixel interpolation and a bilinear filter is used for quarter-pixel interpolation [26, 27, 28]. In HEVC standard, 3 different 8-tap FIR filters are used for both half-pixel and quarter-pixel interpolations [1, 29]. In H.264, 4x4 and 16x16 block sizes are used. However, in HEVC, prediction unit (PU) sizes can be from 4x4 to 64x64. Therefore, HEVC sub-pixel interpolation is more complex than H.264 sub-pixel interpolation [18].

Therefore, in this thesis, a reconfigurable HEVC sub-pixel (half-pixel and quarter-pixel) interpolation hardware for all PU sizes is proposed. The proposed hardware is implemented in Verilog HDL. The Verilog RTL code is verified to work at 100 MHz in a Xilinx Virtex 6 FPGA. The proposed reconfigurability reduces the area and power consumption of HEVC sub-pixel interpolation hardware more than 30%. The proposed hardware, in the worst case, can process 64 quad full HD (2560x1600) video frames per second [25].

An HEVC sub-pixel interpolation hardware only for 4x4 PU size is proposed in [30]. This hardware is slower and has larger area than the hardware proposed in this thesis, because it has restricted reconfigurability. Its power consumption is not reported.

4.1 HEVC Sub-Pixel Interpolation Algorithm

In HEVC standard, 3 different 8-tap FIR filters are used for both half-pixel and quarter-pixel interpolations. These 3 FIR filters type A, type B and type C are shown in (4.1), (4.2), and (4.3), respectively. The shift1 value is determined based on bit depth of the pixel.

$$a_{0,0} = (-A_{-3,0} + 4 * A_{-2,0} - 10 * A_{-1,0} + 58 * A_{0,0} + 17 * A_{1,0} - 5 * A_{2,0} + A_{3,0}) \gg \text{shift1} \quad (4.1)$$

$$b_{0,0} = (-A_{-3,0} + 4 * A_{-2,0} - 11 * A_{-1,0} + 40 * A_{0,0} + 40 * A_{1,0} - 11 * A_{2,0} + 4 * A_{3,0} - A_{4,0}) \gg \text{shift1} \quad (4.2)$$

$$c_{0,0} = (A_{-2,0} - 5 * A_{-1,0} + 17 * A_{0,0} + 58 * A_{1,0} - 10 * A_{2,0} + 4 * A_{3,0} - A_{4,0}) \gg \text{shift1} \quad (4.3)$$

Integer pixels ($A_{x,y}$), half pixels ($a_{x,y}$, $b_{x,y}$, $c_{x,y}$, $d_{x,y}$, $h_{x,y}$, $n_{x,y}$) and quarter pixels ($e_{x,y}$, $f_{x,y}$, $g_{x,y}$, $i_{x,y}$, $j_{x,y}$, $k_{x,y}$, $p_{x,y}$, $q_{x,y}$, $r_{x,y}$) in a PU are shown in Figure 4.1. The half pixels a, b, c are interpolated from nearest integer pixels in horizontal direction, and the half-pixels d, h, n are interpolated from nearest integer pixels in vertical direction. The quarter pixels e, f, g are interpolated from the nearest half pixels a, b, c respectively in horizontal direction using type A filter. The quarter pixels i, j, k are interpolated similarly using type B filter, and the quarter pixels p, q, r are interpolated similarly using type C filter.

HEVC sub-pixel interpolation algorithm used in HEVC decoder calculates the sub-pixels necessary for the given sub-pixel accurate motion vector. If the x fraction and y fraction of the given sub-pixel accurate motion vector are zero, it only performs a shift operation on the integer pixels. If either x fraction or y fraction are zero, it interpolates necessary half-pixels. Otherwise, it interpolates necessary quarter pixels. The necessary sub-pixels for possible x fraction and y fraction values for the PU shown in Figure 4.1 are given in Table 4.1. As shown in Table 4.2, the amounts of

$A_{-1,-1}$				$A_{0,-1}$	$a_{0,-1}$	$b_{0,-1}$	$c_{0,-1}$	$A_{1,-1}$				$A_{2,-1}$
$A_{-1,0}$				$A_{0,0}$	$a_{0,0}$	$b_{0,0}$	$c_{0,0}$	$A_{1,0}$				$A_{2,0}$
$d_{-1,0}$				$d_{0,0}$	$e_{0,0}$	$f_{0,0}$	$g_{0,0}$	$d_{1,0}$				$d_{2,0}$
$h_{-1,0}$				$h_{0,0}$	$i_{0,0}$	$j_{0,0}$	$k_{0,0}$	$h_{1,0}$				$h_{2,0}$
$n_{-1,0}$				$n_{0,0}$	$p_{0,0}$	$q_{0,0}$	$r_{0,0}$	$n_{1,0}$				$n_{2,0}$
$A_{-1,1}$				$A_{0,1}$	$a_{0,1}$	$b_{0,1}$	$c_{0,1}$	$A_{1,1}$				$A_{2,1}$
$A_{-1,1}$				$A_{0,1}$	$a_{0,2}$	$b_{0,2}$	$c_{0,2}$	$A_{1,1}$				$A_{2,1}$

Figure 4.1 Integer, Half and Quarter Pixels

computations performed for interpolating the necessary sub-pixels for different x fraction and y fraction values are different. In the table, w and h represent width and height of the PU respectively.

Table 4.1 Necessary Sub-Pixels for Possible X Fraction and Y Fraction Values

X Fraction	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
Y Fraction	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
predSample	d	h	n	a	e	i	p	b	f	j	q	c	g	k	r

Table 4.2 Amounts of Computations for Sub-Pixel Interpolation

Position	Addition	Multiplication	Shift
a,c,d,n	$6*w*h$	$5*w*h$	$w*h$
b,h	$7*w*h$	$6*w*h$	$w*h$
e,g,p,r	$12*w*h + 36*w$	$10*w*h + 30*w$	$2*w*h + 6*w$
i,k	$13*w*h + 42*w$	$11*w*h + 35*w$	$2*w*h + 7*w$
f,q	$13*w*h + 42*w$	$11*w*h + 36*w$	$2*w*h + 6*w$
j	$14*w*h + 49*w$	$12*w*h + 42*w$	$2*w*h + 7*w$

4.2 Proposed Reconfigurable HEVC Sub-Pixel Interpolation Hardware

The proposed reconfigurable HEVC sub-pixel (half-pixel and quarter-pixel) interpolation hardware for all PU sizes is shown in Figure 4.3. The proposed hardware interpolates the sub-pixels for the luma component of a PU for a given sub-pixel accurate motion vector using integer pixels. Two buffers are used to store integer and half pixels necessary for interpolating the half and quarter pixels. The interpolated pixels are stored in the filtered pixels buffer. These on-chip buffers reduce the required off-chip memory bandwidth and power consumption.

As shown in Figure 4.2, three FIR filters (type A, type B, type C) can be implemented separately and the result can be selected after filtering. In this thesis, this datapath is called the original sub-pixel interpolation datapath.

Since the coefficients of type A and type C filters are symmetric as shown in Table 4.3, they can be implemented in the same datapath by only changing its inputs. However, as shown in Figure 4.4, in the proposed hardware, all three FIR filters (type A, type B, type C) are implemented using a single reconfigurable datapath. This datapath is reconfigured using the x and y fraction information. 8 reconfigurable datapaths are used to interpolate 8 sub-pixels of a PU in parallel. If the PU size is 8, the sub-pixels are interpolated row by row. Otherwise, since the other PU sizes are multiple of 8, the PU is divided into 8×8 blocks, and the blocks are interpolated one by one.

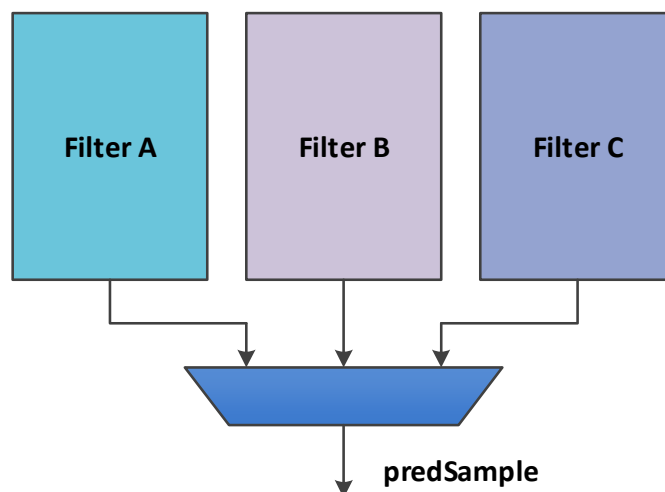


Figure 4.2 Original Sub-Pixel Interpolation Datapath

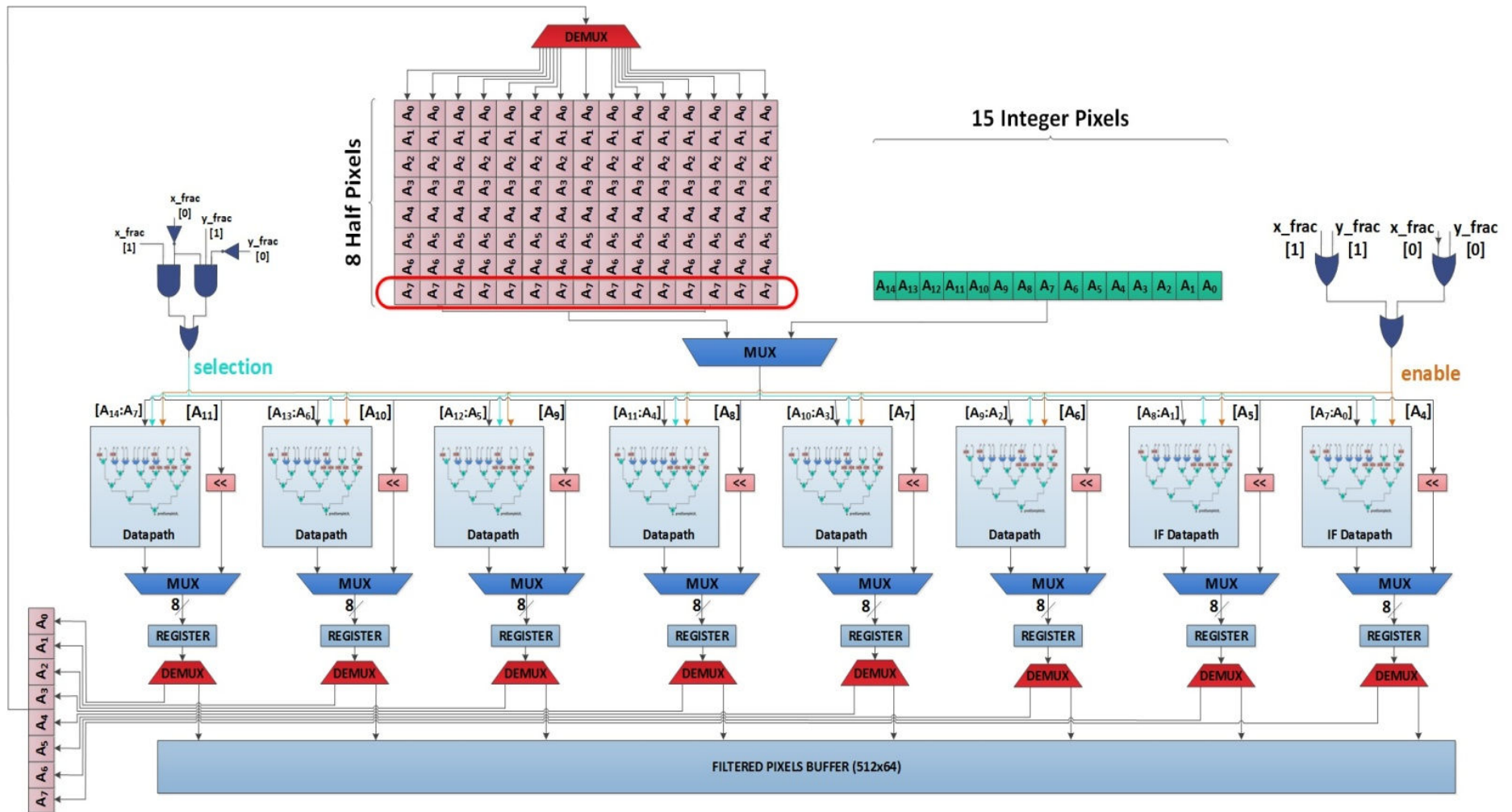


Figure 4.3 HEVC Sub-Pixel Interpolation Hardware

Table 4.3 FIR Filter Coefficients

Type	Coefficients
A	{-1,4,-10,58,17,-5,1}
B	{-1,4,-11,40,40,-11,4,-1}
C	{1,-5,17,58,-10,4,-1}

The proposed hardware, in the worst case, interpolates the sub-pixels in an 8x8 PU (64 pixels) in 24 clock cycles. If x fraction and y fraction of the given sub-pixel accurate motion vector are zero, sub-pixel interpolation datapaths are disabled and only integer pixels are shifted. Otherwise, first, 15 integer pixels are loaded into integer pixels buffer in one clock cycle. Then, 8x15 half pixels necessary for interpolating quarter pixels are interpolated in 15 clock cycles, and 64 quarter pixels are interpolated in 8 clock cycles.

Both the proposed reconfigurable interpolation hardware and the same interpolation hardware without reconfigurability (original sub-pixel interpolation hardware) are implemented in Verilog HDL. The hardware implementations are verified with RTL simulations using Mentor Graphics Questa. The RTL simulation results matched the results of a software model of HEVC sub-pixel interpolation algorithm.

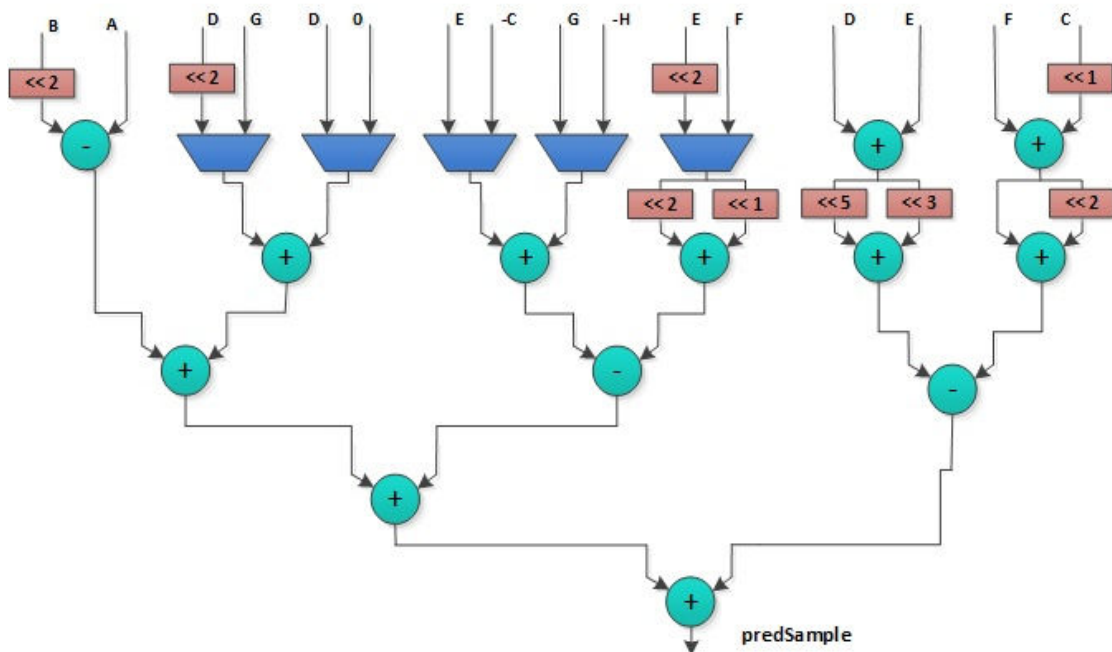


Figure 4.4 Reconfigurable Sub-Pixel Interpolation Datapath

The Verilog RTL codes of both interpolation hardware are mapped to a Xilinx XC6VLX75T FF784 FPGA with speed grade 3 using Xilinx ISE 13.4. Original FPGA implementation uses 3005 LUTs, 1224 DFFs and 2 BRAMs. Reconfigurable FPGA implementation uses 1890 LUTs, 1224 DFFs, and 2 BRAMs. Both FPGA implementations are verified to work at 100 MHz by post place & route simulations. Both original and reconfigurable FPGA implementations, in the worst-case, can process 64 quad full HD (2560x1600) video frames per second.

The power consumptions of both FPGA implementations are estimated using Xilinx XPower Analyzer tool for Tennis (1920x1080), Basketball Drive (1920x1080), Cactus (1920x1080) and BQTerrace (1920x1080) video frames [14]. These power consumptions are shown in Table 4.4. As shown in this table, the proposed reconfigurability reduced the power consumption of original HEVC sub-pixel interpolation hardware more than 30%.

In order to estimate the power consumption of a sub-pixel interpolation hardware, timing simulation of its placed and routed netlist is done at 100 MHz using Mentor Graphics Questa for HEVC video decoding of one frame of each video sequence. The signal activities of these timing simulations are stored in VCD files, and these VCD files are used for estimating the power consumption of that sub-pixel interpolation hardware using Xilinx XPower Analyzer tool. Since sub-pixel interpolation hardware will be used as part of a HEVC encoder or decoder, only internal power consumption is considered, and input and output power consumptions are ignored. Therefore, the power consumption of sub-pixel interpolation hardware can be divided into four main categories; clock power, logic power, signal power and BRAM power.

The Verilog RTL code of the proposed reconfigurable HEVC sub-pixel interpolation hardware is also synthesized and place & routed to Synopsys 90nm standard cell library. The gate count of resulting ASIC implementation is calculated as 10.5k, excluding on-chip memories, based on NAND (2x1) gate area.

An HEVC sub-pixel interpolation hardware only for 4x4 PU size is proposed in [30]. This hardware has restricted reconfigurability. It finishes sub-pixel interpolations of a 16x16 CU in 352 clock cycles. But, the hardware proposed in this thesis finishes sub-pixel interpolations of a 16x16 CU in 96 clock cycles. It is implemented using SMIC 90nm standard cell library, and its gate count is reported as 19.6k gates.

Therefore, it also has larger area than the hardware proposed in this thesis. Its power consumption is not reported.

Table 4.4 Power Consumption Reductions for 1920x1080 Video Frames

	Tennis		Basketball Drive		Cactus		BQ Terrace	
	Org.	Low Power	Org.	Low Power	Org.	Low Power	Org.	Low Power
Clock(mW)	16	18	16	18	15	17	14	16
Logic(mW)	75	35	74	35	80	38	78	37
Signal(mW)	69	53	68	54	76	60	75	59
BRAM(mW)	5	5	5	5	4	4	3	3
Total Time(ms)	2.682		2.835		2.659		3.729	
Total Power(mW)	165	111	163	112	175	119	170	115
Energy(uJ)	442.53	297.70	462.11	317.52	465.33	316.42	633.93	428.83
Power Red.	32.73%		31.29%		32.00%		32.35%	

CHAPTER V

CONCLUSIONS AND FUTURE WORK

In this thesis, we proposed novel computational complexity and energy reduction techniques for intra prediction algorithm used in HEVC video encoder and decoder. We quantified the computation reductions achieved by these techniques using HEVC HM reference software encoder. We designed efficient hardware architectures for these video compression algorithms used in HEVC. We also designed a reconfigurable sub-pixel interpolation hardware for both HEVC encoder and decoder. We implemented these hardware architectures in Verilog HDL. We mapped the Verilog RTL codes to a Xilinx Virtex 6 FPGA and estimated their power consumptions on this FPGA using Xilinx XPower Analyzer tool. The proposed techniques significantly reduced the energy consumptions of these FPGA implementations in some cases with no PSNR loss and in some cases with very small PSNR loss.

As future work, DC and planar modes, and all PU sizes can be added to the proposed HEVC intra prediction hardware. An HEVC video encoder can be implemented by implementing the parts of an HEVC video encoder which are not implemented in this thesis and by integrating them with the ones implemented in this thesis.

BIBLIOGRAPHY

- [1] B. Bross, W.J. Han, J.R. Ohm, G.J. Sullivan, Y.K. Wang, and T. Wiegand, "High Efficiency Video Coding (HEVC) Text Specification Draft 10", *JCTVC-L1003*, Feb. 2013.
- [2] G.J.Sullivan, J.R. Ohm, W.J. Han, T. Wiegand, " Overview of the High Efficiency Video Coding (HEVC) Standard,"*IEEE Trans. on Circuits and Systems for Video Technology*, vol.22, no.12, pp.1649-1668, Dec. 2012.
- [3] F. Bossen, B. Bross, K. Suhring and D. Flynn, "HEVC Complexity and Implementation Analysis ", *IEEE Trans. on Circuits and Systems for Video Technology*, vol.22, no.12, pp.1685-1696, Dec. 2012.
- [4] J. Vanne, M. Viitanen, T.D. Hämäläinen and A. Hallapuro, "Comparative Rate-Distortion-Complexity Analysis of HEVC and AVC Video Codecs", *IEEE Trans. on Circuits and Systems for Video Technology*, vol.22, no.12, pp.1885-1898, Dec. 2012.
- [5] C.K. Huang, L.C. Wu, H.T. Huang, T.H. Sheng, L.L. Youn, "A Low-Power High-Performance H.264/AVC Intra-Frame Encoder for 1080p HD Video", *IEEE Trans. on Very Large Scale Integration Systems*, vol.19, no.6, pp.925-938, June 2011.
- [6] G. J. Sullivan, G. Bjøntegaard, and A. Luthra T. Wiegand, "Overview of the H.264/AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.
- [7] J. Lainema, F. Bossen, W.J. Han, J. Min and K. Ugur, "Intra Coding of the HEVC Standard", *IEEE Trans. on Circuits and Systems for Video Technology*, vol.22, no.12, pp.1792-1801, Dec. 2012.

- [8] M. Parlak, Y. Adibelli, I. Hamzaoglu, “A Novel Computational Complexity and Power Reduction Technique for H.264 Intra Prediction”, *IEEE Trans. on Consumer Electronics*, vol.54, no. 4, pp. 2006 – 2014, Nov. 2008.
- [9] Y. Adibelli, M. Parlak, I. Hamzaoglu, “Pixel Similarity Based Computation and Power Reduction Technique for H.264 Intra Prediction”, *IEEE Trans. on Consumer Electronics*, vol.56, no. 2, pp. 1079-1087, May 2010.
- [10] Y. Adibelli, M. Parlak, I. Hamzaoglu, “Computation and Power Reduction Techniques for H.264 Intra Prediction”, *Microprocessors and Microsystems: Embedded Hardware Design*, vol. 36, issue 3, May 2012.
- [11] E. Kalali, Y. Adibelli, I. Hamzaoglu, “A High Performance and Low Energy Intra Prediction Hardware for High Efficiency Video Coding”, *Int. Conference on Field Programmable Logic and Applications*, Aug. 2012.
- [12] K. McCann, B. Bross, I.K. Kim, S. Sekiguchi, and W.J. Han, “High Efficiency Video Coding (HEVC) Test Model 5 Encoder Description”, *JCTVC-G1102*, Nov. 2011.
- [13] F. Li, G. Shi, F. Wu, “An Efficient VLSI Architecture for 4x4 Intra Prediction in High Efficiency Video Coding Standard”, *IEEE Int. Conf. on Image Processing*, Sep. 2011.
- [14] F. Bossen, “Common test conditions and software reference configurations”, *JCTVC-G1200*, Nov. 2011.
- [15] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves”, *13th Video Coding Experts Group Meeting*, 2001.
- [16] E. Kalali, Y. Adibelli, I. Hamzaoglu, “A High Performance and Low Energy Intra Prediction Hardware for HEVC Video Decoding”, *Conference on Design and Architectures for Signal and Image Processing*, Oct. 2012.
- [17] S. Park, J. Park, and B. Joen, “Report on the evaluation of HM versus JM,” *JCTVC-D181*, Jan. 2011.
- [18] I. Richardson, “The H.264 Advanced Video Compression Standard”, *Wiley*, 2010.
- [19] G.V. Wallendael, S.V. Leuven, J.D. Cock, P. Lambert, R. V. Walle, J. Barbarien, and A. Munteanu, “Improved Intra Mode Signaling for HEVC,” *IEEE Int. Conf. on Multimedia and Expo*, July 2011.
- [20] E. Sahin, I. Hamzaoglu, “An Efficient Hardware Architecture for H.264 Intra Prediction Algorithm”, *Design, Automation and Test in Europe Conference*, April 2007.
- [21] Y. Lai, T. Liu, Y. Li, C. Lee, “Design of An Intra Predictor with Data Reuse for High-Profile H.264 Applications”, *IEEE ISCAS*, May 2009.

- [22] Y. Adibelli, M. Parlak, I. Hamzaoglu, "A Computation and Power Reduction Technique for H.264 Intra Prediction", *Euromicro Conf. on Digital Systems Design*, Sep. 2010.
- [23] M Alvarez-Mesa, C.C. Chi, B. Juurlink, V. George, T. Schierl, "Parallel Video Decoding In The Emerging HEVC Standard", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 2012.
- [24] Y.J. Ahn, W.J. Han, D. G. Sim, "Study of Decoder Complexity for HEVC and AVC Standarts Based on Tool-by-Tool Comparison", *Proc. SPIE Applications of Digital Image Processing XXXV*, vol. 8499, Aug. 2012.
- [25] E. Kalali, Y. Adibelli, I. Hamzaoglu, "A Reconfigurable HEVC Sub-Pixel Interpolation Hardware", *IEEE Int. Conf. on Consumer Electronics - Berlin*, Sep. 2013.
- [26] J. Kim, J. Kim, K. Yoo, and K. Lee, "Analysis and Complexity Reduction of High Efficiency Video Coding for Low-Delay Communication", *IEEE Int. Conf. on Consumer Electronics – Berlin*, Sep. 2012.
- [27] S. Yalcin, I. Hamzaoglu, "A High Performance Hardware Architecture for Half-Pixel Accurate H.264 Motion Estimation", *14th IFIP Int. Conference on VLSI-SoC*, Oct. 2006.
- [28] S. Oktem, I. Hamzaoglu, "An Efficient Hardware Architecture for Quarter-Pixel Accurate H.264 Motion Estimation", *10th Euromicro Conference on Digital System Design*, Aug. 2007.
- [29] M. T. Pourazad, C. Doutre, M. Azimi, P. Nasiopoulos, "HEVC: The New Gold Standard for Video Compression", *IEEE Consumer Electronics Magazine*, July 2012.
- [30] Z. Guo, D. Zhou, S. Goto, "An Optimized MC Interpolation Architecture for HEVC", *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 1117-1120, March 2012.