

FPGA Implementation of HEVC Intra Prediction Using High-Level Synthesis

Ercan Kalali, Ilker Hamzaoglu

Faculty of Engineering and Natural Sciences, Sabanci University
34956 Tuzla, Istanbul, Turkey
{ercankalali, hamzaoglu}@sabanciuniv.edu

Abstract—Intra prediction algorithm in the recently developed High Efficiency Video Coding (HEVC) standard has very high computational complexity. High-level synthesis (HLS) tools are started to be successfully used for FPGA implementations of digital signal processing algorithms. Therefore, in this paper, the first FPGA implementation of HEVC intra prediction algorithm using a HLS tool in the literature is proposed. The proposed HEVC intra prediction hardware, in the worst case, can process 35 full HD (1920x1080) video frames per second. Using HLS tool significantly reduced the FPGA development time. Therefore, HLS tools can be used for FPGA implementation of HEVC video encoder.

Keywords—HEVC, Intra Prediction, HLS, FPGA.

I. INTRODUCTION

Joint collaborative team on video coding (JCT-VC) recently developed a new video compression standard called High Efficiency Video Coding (HEVC) [1]-[4]. It has 50% better video compression efficiency than H.264 standard. Intra prediction algorithm predicts the pixels of a block from the pixels of its already coded and reconstructed neighboring blocks. In HEVC, for the luminance component of a frame, intra prediction unit (PU) sizes can be from 4x4 up to 32x32 and number of intra prediction modes for a PU can be up to 35 [2].

In this paper, the first high-level synthesis (HLS) implementation of HEVC intra prediction algorithm in the literature is proposed. The proposed HEVC intra prediction hardware is implemented on Xilinx FPGAs using Xilinx Vivado HLS tool. HLS tools accept their inputs in different formats [5]. Xilinx Vivado HLS tool takes C or C++ codes as input, and generates Verilog or VHDL codes. The C codes given as input to Xilinx Vivado HLS tool are developed based on the HEVC intra prediction software implementation in the HEVC reference software video encoder (HM) version 15 [6].

The Verilog RTL codes are synthesized and mapped to a Xilinx XC6VLX130T FF1156 FPGA with speed grade 3 using Xilinx ISE 14.7. The proposed FPGA implementation of HEVC intra prediction using HLS, in the worst case, can process 35 full HD (FHD) (1920x1080) video frames per second.

A few HLS implementations for HEVC video compression standard are proposed in the literature [7]-[9]. A few HLS

implementations for H.264 video compression standard are proposed in the literature [10]. There are a few HLS implementations based on MPEG reconfigurable video coding [11], [12]. There are several HLS implementations for image and video processing algorithms such as sorting in the median filter [13]-[16].

In Section III, the HEVC intra prediction HLS implementation proposed in this paper is compared with the handwritten HEVC intra prediction hardware proposed in the literature [2], [17]-[19].

The rest of the paper is organized as follows. HEVC intra prediction algorithm is explained in Section II. In Section III, the proposed HLS implementation is explained, and the implementation results are given. Finally, Section IV presents the conclusions.

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

A. HEVC Intra Angular Prediction

In HEVC intra angular 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.

After the reference main array is determined, the index to this array and the coefficient of pixels are calculated as shown in (1a) and (1b), respectively.

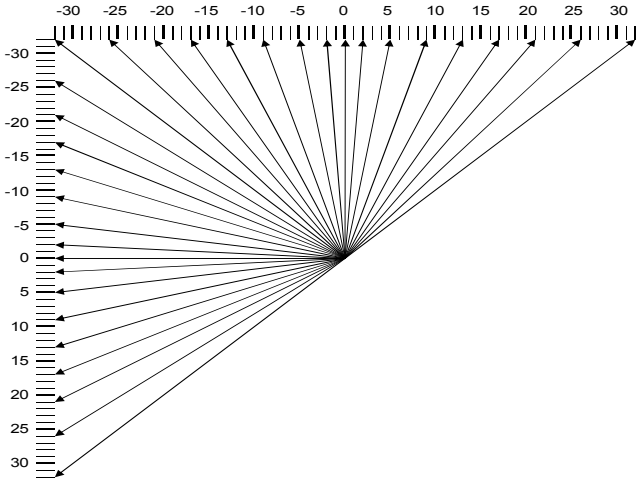


Fig. 1. HEVC Intra Prediction Mode Directions

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

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

If $iFact$ is equal to 0, neighboring pixels are copied directly to predicted pixels. Otherwise, predicted pixels are calculated as shown in (2).

$$Angular[x,y] = ((32-iFact)*refMain[x+iIdx+1] + iFact*refMain[x+iIdx+2]+16) \gg 5 \quad (2)$$

B. HEVC Intra DC Prediction

In HEVC intra DC prediction algorithm, first, average of top and left neighboring pixels ($DCVal$) is calculated. The pixels in a PU except boundary pixels $(0,0)$, $(x,0)$, $(0,y)$ are predicted as $DCVal$. Boundary pixels are predicted as weighted average of $DCVal$ and reference pixels as shown in (3a), (3b) and (3c).

$$DC(0,0) = (P(-1,0)+2*DCVal+P(0,-1)+2) \gg 2 \quad (3a)$$

$$DC(x,0) = (P(x,-1)+3*DCVal+2) \gg 2 \quad (3b)$$

$$DC(0,y) = (P(-1,y)+3*DCVal+2) \gg 2 \quad (3c)$$

C. HEVC Intra Planar Prediction

HEVC intra planar prediction algorithm predicts the pixels in a PU as weighted average of four reference pixels as shown in (4). Coefficients and indexes used in this equation are determined by PU size and pixel location. Two of the reference pixels are selected from top neighboring pixels, and the other two are selected from left neighboring pixels.

$$Planar[x,y] = ((PU-1-x)*P(-1,y) + (x+1)*P(PU,-1) + (PU-1-y)*P(x,-1) + (y+1)*P(-1,PU)+PU) \gg (\log_2(PU) + 1) \quad (4)$$

III. PROPOSED HLS IMPLEMENTATION

The proposed HLS implementation of HEVC intra prediction is shown in Fig. 2. The proposed HLS implementation is synthesized to Verilog RTL using Xilinx Vivado HLS tool. The C codes given as input to Xilinx Vivado HLS tool are developed based on the HEVC intra prediction software implementation in the HEVC reference software video encoder (HM) version 15 [6].

In the proposed HLS implementation, angular, DC and planar prediction modes are implemented for all PU sizes (4x4, 8x8, 16x16 and 32x32). Multiplications with constants are implemented using addition and shift operations for angular and DC prediction modes. However, planar prediction mode uses multiplication operations. 16 different arrays are used to store neighboring pixels. After these neighboring pixels are loaded in 64 clock cycles, three different functions are used to calculate prediction equations for all PU sizes and prediction modes. Angular prediction function can calculate 32 pixels per clock cycle, planar prediction function can calculate 4 pixels per clock cycle, and DC prediction function can calculate 1 pixel per clock cycle in the proposed HLS implementation.

Verilog RTL codes generated by Xilinx Vivado HLS tool for this HLS implementation are verified with RTL simulations. RTL simulation results matched the results of HEVC intra prediction software implementation in the HEVC reference software video encoder (HM) version 15 [6].

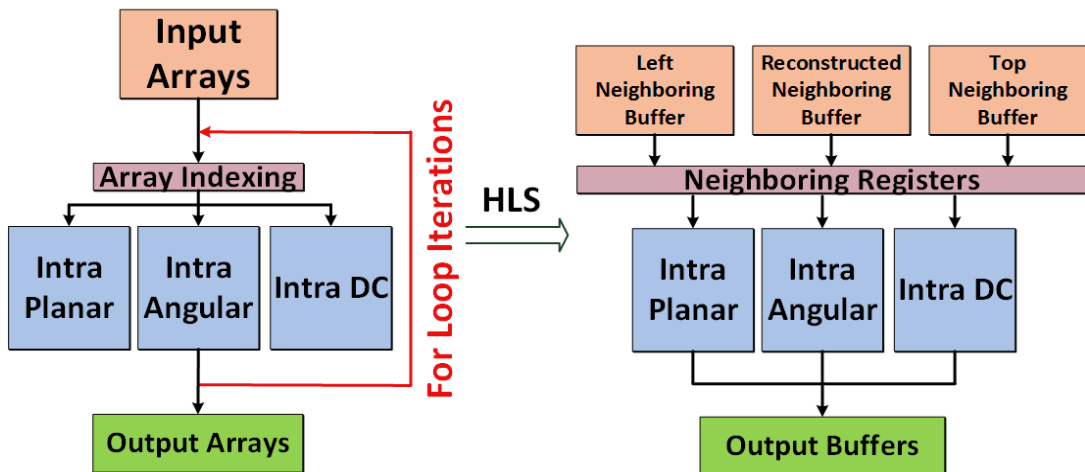


Fig. 2. HEVC Intra Prediction HLS Implementation

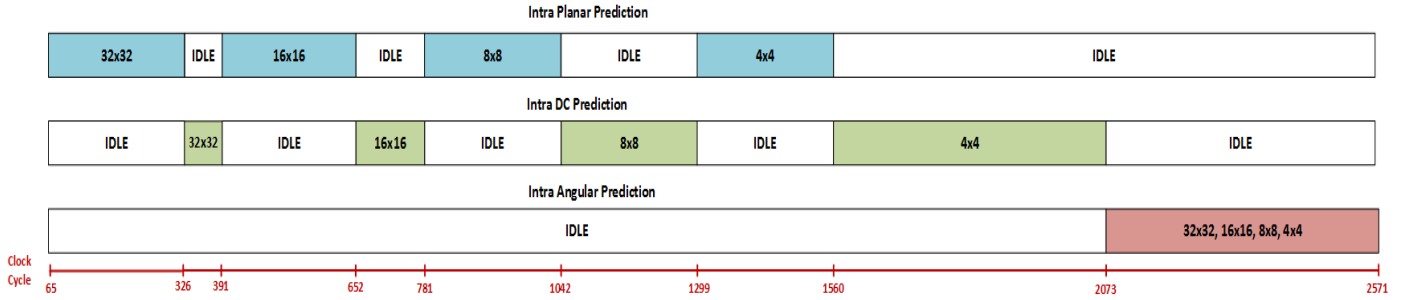


Fig. 3. Scheduling of HEVC Intra Prediction HLS Implementation

TABLE I. HLS IMPLEMENTATION RESULTS

Optimizations	Slice	LUT	DFF	BRAM	DSP48	Freq. (MHz)	Clock Cycles	Fps (1920x1080)
NOOPT	1357	3636	2016	0	17	172	13602	6
ALC(M20)	1381	3665	1963	0	13	182	12578	7
PIPE	1737	4108	2046	0	17	182	2571	35
RES(BRAM)	1120	3147	1638	8	17	182	13602	7
ALC(M20)_RES(BRAM)_PIPE	1473	3525	1686	8	13	160	2571	31
ALC(M20)_APAR_RES(BRAM)_PIPE_BIT	1447	3634	1666	8	12	182	2571	35

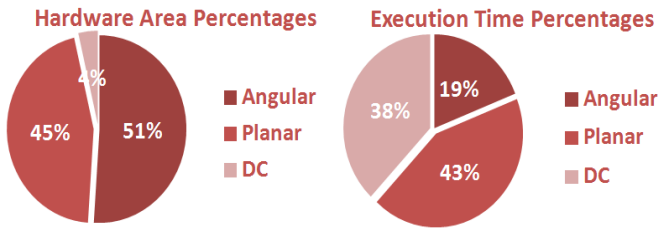


Fig. 4. Hardware Area and Execution Time Percentages

The Verilog RTL codes are synthesized and mapped to a Xilinx XC6VLX130T FF1156 FPGA with speed grade 3 using Xilinx ISE 14.7. The FPGA implementations are verified with post place and route simulations. We performed manual loop unrolling in the proposed HLS implementation to increase its performance. We also used several optimizations offered by Xilinx Vivado HLS tool to increase the performance and decrease the area of the proposed HLS implementation [20].

Allocation (ALC) directive is used to specify the maximum number of resources that can be used in hardware. It forces the HLS tool to perform resource sharing. It therefore decreases the hardware area. In the proposed HLS implementation, ALC is used for multiplication operations. Pipeline (PIPE) directive performs pipelining to increase the performance. It is used in the proposed HLS implementation.

Resource (RES) directive is used to specify which resource will be used to implement a variable such as an array, arithmetic operation or function argument. In the proposed

HLS implementation, it is used to store the pixels in Reconstructed Neighboring Buffer into BRAMs.

Array partition (APAR) directive partitions the large arrays into multiple smaller arrays or individual registers for parallel data accesses. In the proposed HLS implementation, it is used to partition the arrays that store neighboring pixels to increase performance. Xilinx Vivado HLS tool provides a specific library for designing bit-accurate (BIT) models in C codes. In the proposed HLS implementation, bit accurate model is used to decrease adder bit widths and therefore hardware area.

The FPGA implementation results for the HLS implementation are given in Table I. In the HLS implementation, in the C codes, multiplications with constants are implemented using addition and shift operations for DC and angular prediction modes. However, planar prediction mode uses multiplication operations. These multiplication operations are mapped to DSP48 blocks in RTL synthesis. In the table, M shows the number of multipliers used in the ALC directive.

Allocation (ALC), array partition (APAR), resource (RES), pipeline (PIPE) and BIT accurate model are used in the best HEVC intra prediction HLS implementation proposed in this paper (ALC(M20)_APAR_RES(BRAM)_PIPE_BIT). Scheduling of this HLS implementation is shown in Fig. 3. Hardware area and execution time percentages of this HLS implementation are shown in Fig. 4. This HLS implementation is compared with the handwritten HEVC intra prediction hardware implementations proposed in the literature [2], [17]-[19].

TABLE II. HEVC INTRA PREDICTION HARDWARE COMPARISON

	FPGA	DFP	LUT	Freq. (MHz)	Fps	PU
[2]	Xilinx Virtex 6	849	2381	150	30 FHD	4x4, 8x8
[17]	Altera Aria II	---	20496	100	15 FHD	All
[18]	Altera EP2AGX	6934	13409	162	---	All
[19]	Xilinx Virtex 6	140 K	185 K	213	30 QFHD	All
Prop.	Xilinx Virtex 6	1666	3634	182	35 FHD	All

As shown in Table II, since the handwritten HEVC intra prediction hardware proposed in [2] implements only angular prediction modes for 4x4 and 8x8 PUs, it has smaller area than the proposed HLS implementation. The handwritten HEVC intra prediction hardware proposed in [17], [18] have lower performance and larger area than the proposed HLS implementation. The handwritten HEVC intra prediction hardware proposed in [19] has higher performance (30 Quad full HD) than the proposed HLS implementation, but it has much larger area.

IV. CONCLUSIONS

In this paper, the first FPGA implementation of HEVC intra prediction algorithm using a HLS tool in the literature is proposed. The proposed HEVC intra prediction hardware is implemented on Xilinx FPGAs using Xilinx Vivado HLS tool. Using HLS tool significantly reduced the FPGA development time. The implementation results show that the proposed HEVC intra prediction FPGA implementation, in the worst case, can process 35 full HD (1920x1080) video frames per second. Therefore, HLS tools can be used for FPGA implementation of HEVC video encoder.

ACKNOWLEDGMENT

This research was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) under the contract 115E290.

REFERENCES

- [1] High Efficiency Video Coding, ITU-T Rec. H.265 and ISO/IEC 23008-2 (HEVC), ITU-T and ISO/IEC, Apr. 2013.
- [2] 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.
- [3] E. Kalali, E. Ozcan, O. M. Yalcinkaya, I Hamzaoglu, "A Low Energy HEVC Inverse Transform Hardware", IEEE Trans. on Consumer Electronics, vol. 60, no. 4, pp. 754-761, Nov. 2014.
- [4] E. Kalali, Y. Adibelli, I. Hamzaoglu, "A Reconfigurable HEVC Sub-Pixel Interpolation Hardware", IEEE Int. Conference on Consumer Electronics - Berlin, Sept. 2013.
- [5] W. Meeus, K. V. Beeck, T. Goedeme, J. Meel, D. Stroobandt, "An Overview of Today's High-Level Synthesis Tools," Springer Design

- Automation for Embedded Systems, vol. 16, no. 3, pp. 31-51, Sept. 2012.
- [6] K. McCann, B. Bross, W.J. Han, I.K. Kim, K. Sugimoto, G. J. Sullivan, "High Efficiency Video Coding (HEVC) Test Model (HM) 15 Encoder Description", JCTVC-Q1002, June 2014.
- [7] E. Kalali, I. Hamzaoglu, "FPGA Implementations of HEVC Inverse DCT Using High-Level Synthesis," Conf. on Design and Architectures for Signal and Image Processing, Sept. 2015.
- [8] F. A. Ghani, E. Kalali, I. Hamzaoglu, "FPGA Implementations of HEVC Sub-Pixel Interpolation Using High-Level Synthesis," Int. Conf. on Design and Technology of Integrated Systems, April 2016.
- [9] P. Sjoval, J. Virtanen, J. Vanne, T. D. Hamalainen, "High-Level Synthesis Design Flow for HEVC Intra Encoder on SoC-FPGA," Euromicro Conf. on Digital System Design, pp. 49-56, Aug. 2015.
- [10] S. Kim, H. Kim, T. Chung, J-G. Kim, "Design of H.264 Video Encoder with C to RTL Design Tool," Int. SoC Design Conference, pp. 171-174, Nov. 2012.
- [11] S. S. Bhattacharyya, J. Eker, J. W. Janneck, C. Lucarz, M. Mattavelli, M. Raulet, "Overview of the MPEG Reconfigurable Video," Springer Journal of Signal Processing Systems, vol. 63, no. 2, pp. 251-263, May 2011.
- [12] J. F. Nezan, N. Siret, M. Wipliez, F. Palumbo, L. Raffo, "Multi-purpose systems : A novel dataflow-based generation and mapping strategy," IEEE Int. Symposium on Circuits and Systems (ISCAS), pp. 3073-3076, May 2012.
- [13] H. Ye, L. Lacassagne, D. Etiemble, L. Cabaret, J. Falcou, A. Romero, O. Florent, "Impact of high level transforms on high level synthesis for motion detection algorithm," Conf. on Design and Architectures for Signal and Image Processing, Oct. 2012.
- [14] G. Schewior, C. Zahl, H. Blume, S. Wonneberger, J. Effertz, "HLS-based FPGA implementation of a predictive block-based motion estimation algorithm - A field report," Conf. on Design and Architectures for Signal and Image Processing, Oct. 2014.
- [15] O. A. Abella, G. Ndu, N. Sonmez, M. Ghasempour, A. Armejach, J. Navaridas, W. Song, J. Mawer, A. Cristal, M. Lujan, "An empirical evaluation of high-level synthesis languages and tools for database acceleration," Int. Conf. on Field Programmable Logic and Applications, Sept. 2014.
- [16] M. Schmid, N. Apelt, F. Hanning, J. Teich, "An Image Processing Library for C-based High-Level Synthesis," Int. Conf. on Field Programmable Logic and Applications, Sept. 2014.
- [17] A. Abramowski, G. Pastuszak, "A Novel Intra Prediction Architecture for the Hardware HEVC Encoder," Euromicro Conf. Digital System Design, pp. 429-436, Sept. 2013.
- [18] M. U. K. Khan, M. Shafique, M. Grellert, J. Henkel, "Hardware-Software Collaborative Complexity Reduction Scheme for The Emerging HEVC Intra Encoder," Design, Automation and Test in Europe (DATE) Conference, pp. 125-128, March 2013.
- [19] F. Amish, E. B. Bourennan, "A Novel Hardware Accelerator for The HEVC Intra Prediction," IEEE Int. Conf. on New Circuits and Systems, June 2015.
- [20] UG902, "Vivado Design Suite User Guide: High-Level Synthesis," May 2014.