MV – HEVC Chip Design Using High Level Synthesis

Hye Bin Ha[1], Sang Un Park[2], Wei Liu and Yong Beom Cho[a]
Department of Electrical Engineering, Konkuk University
E-mail: [1]hb941224@konkuk.ac.kr, [2]kos8108@konkuk.ac.kr

***Abstract* - This study proposed a system that MV-HEVC design using High-Level Synthesis (HLS). There are several improvements in MV-HEVC Design. First of all, using HLS tool enables designers to quickly verify feasibility of different hardware-software boundaries. The system uses more than Mb size memory even in very small environment such as MV-HEVC. Instead of using memory inside the chip, it uses external memory, especially memory such as DDR1, which can operates at low speeds. IO is not specially designed for memory use, but to allow memory access using GPIO. Low power delay is implemented to control skew between data and memory control that can operate at low speed. The implementation of the algorithm made the chip through the Samsung 65nm foundry process. In conclusion, we designed hardware of MV-HEVC for real-time coding of multi-view video. And the design improves and optimizes the coding speed through integrated design of SW / HW using SoC. And also, we designed a system that can use more than Mb of memory at low speeds, such as MV-HEVC environments.**

***Keywords*—FPGA, HLS, MPW, MV-HEVC**

I. INTRODUCTION

Nowadays, the video data is increasing due to the increase of the resolution. Therefore, the performance of the processor for processing video data is also increasing. Especially, in the case of multi view, more video data processing capability is required. Therefore, it is impossible to process with software. So, the dedicated chip supporting multi-view is needed. In addition, because of the size of the video data, a device for transmitting high-speed data is required together. In addition, Coding multiple images using multi-view image causes a decrease in coding speed in proportion to the number of images. To solve this, the speed of coding of images is reduced through parallelization and differentiation of prediction. Because there are physical limitations, it implements the hardware of MV-HEVC for real-time coding of multi-view video and improves and optimizes the coding speed through integrated design of SW / HW using SoC. Research is needed to process multiple images. In order to overcome these limitations, MV-HEVC implementation using FPGA was performed by utilizing Vivado High-Level Synthesis (HLS) Tool.

a. Corresponding author; ybcho@konkuk.ac.kr

Manuscript Received Jun. 11, 2020, Revised Jul. 15, 2020, Accepted Jul. 15, 2020

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/bync/3.0) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

II. DESIGN FLOW

A. Overall Design Flow

Figure 1 shows overall MV-HEVC for FPGA design flow. HTM 16.0, the standard SW model of MV-HEVC, is designed with MV-HEVC decoder using Xilinx's Vivado HLS.Optimized HW was designed by adding necessary pragmas in the process of converting HTM 16.0 code to HW in the Vivado HLS compiler. As a result, RTL design of about 5 million gates was obtained. We have created a test environment that can verify that the design results actually work on the FPGA. After verification, the modified and complemented design is put on the actual MPW chip and tested.

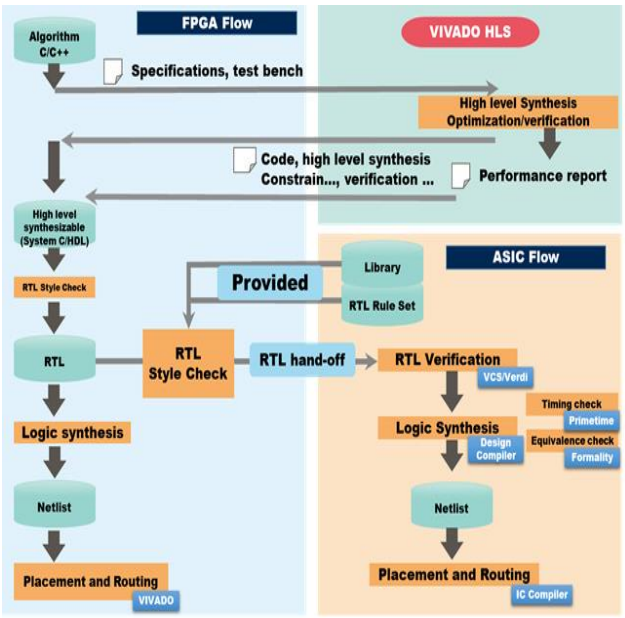


Fig. 1. Design Flow

B. MV-HEVC

MV-HEVC is a quick extension of HEVC, which was established in July 2012 by the Joint Group of 3D Video Coding Extensions Development (JVT-3V) to further improve coding efficiency [1]. MV-HEVC follows the same design principles of MVC and makes use of the redundancy between different views of the same scene to improve compression efficiency.

MV HEVC is a video coding technique using Multiview. As shown in Figure 2, MV HEVC has multiple sequences of information unlike HEVC, so it is not only necessary to

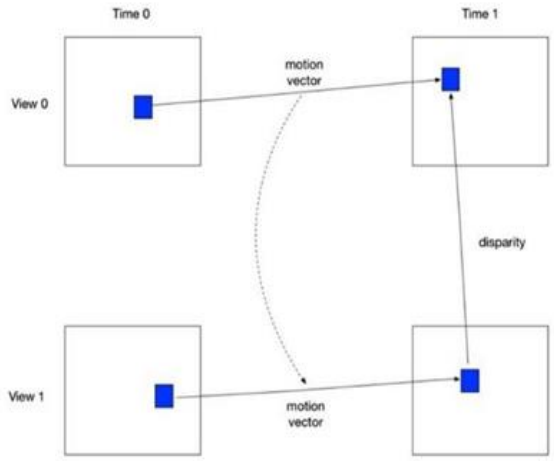


Fig. 2. Motion estimation of MV-HEVC

estimate motion of the time axis, but also essential to estimating the motion of the space axis. Since motion estimates of the time-space axis affect the effect of compression coding, the system's throughput should be increased through the optimal study of the method of prediction between frames and parallelization of structures requiring many computations, such as motion estimation DCT.

Figure 2 shows the prediction method between two frames of point image coding, and Figure 3(a) shows the coding method using only the prediction between frames on a domain based on the existing H.264/AVC and H.265/HEVC, which is based on the simulcast determination [2]. Figure 3(b) is the inter feedback of point image coding, a method for increasing compression efficiency through prediction between frames on the time axis as well as between adjacent cameras. A design is needed to enable the selection of images from the receiver side through transmission of multiple images at the same time, generating one signal flow for a number of images in the encoding process [3].

Figure 4 shows entire MV-HEVC SoC Design including Hardware Architecture. Arm Neon method optimizes Header Parsing, Video Buffer, CABAC Entropy Decoder, and Inverse Transform. Decoder for Multiview High Efficiency Video Coding is implemented using these related function block. Memory is optimized by array_partition module of HLS. It makes Memory read/write function be fatster.

C. Design Process in Vivado HLS

Use of HLS tools is particularly attractive in designs that use system on a chip device. The heterogeneous nature of resources available in those devices requires a methodology that enables designers to quickly verify feasibility of different hardware-software boundaries [4]. IDCT synthesis result between Vivado HLS and Handwritten Verilog is shown in Table I. When hardware description languages are used, the implementation of even simple hardware accelerators can be very time-consuming and require the designer to express the implemented algorithm at a very low, hardware level of abstraction [5][6].

Vivado HLS is a high-level synthesis tool available for Xilinx FPGAs and Zynq SoC devices that can synthesize Verilog and VHDL RTL models from C, C++ and SystemC code.HW is implemented using Verilog HDL converted through the Vivado HLS compiler. Then, performance verification was performed on the FPGA. It was confirmed that the generated HW uses the resources of the FPGA as shown in the Table II. In addition, it was verified that the design can be implemented on an FPGA. Algorithms through optimization of unnecessary parts and codes of MV-HEVC reference code (HTM16.0). In the process of improving, the coding performance of the image is improved.

Optimized MV-HEVC code is divided into functions and suitable for HW and SW as follows. Divided into functions and implemented SoC through Co-Design using SDSoC. Sample Adaptive Offset (SAO), Motion Compensation (MC), Inverse Discrete Consine (IDTC) HW / SW Co-design compared to pure SW implementation through HW implementation. Performance improved by about 1.5 times.

The results synthesized from SDSoC were verified for each cycle in HLS and operated in the correct cycle and verified.

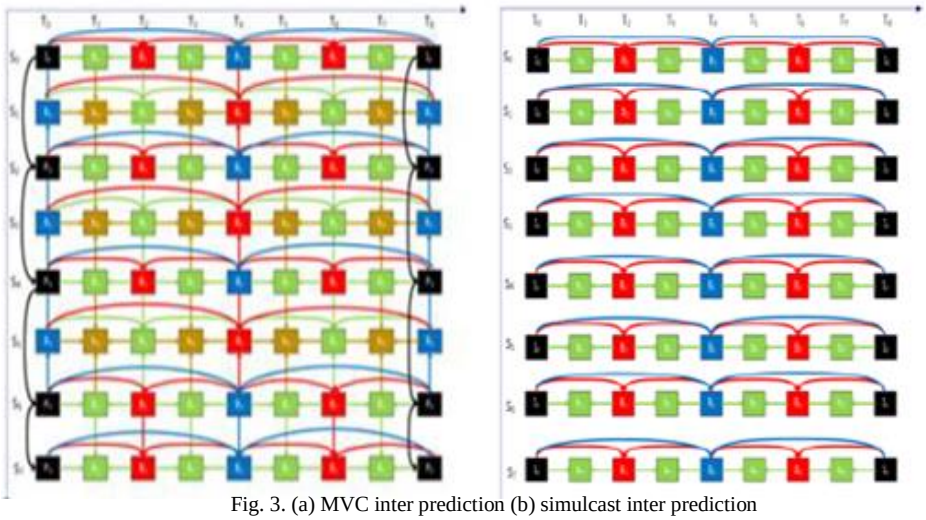


Fig. 3. (a) MVC inter prediction (b) simulcast inter prediction

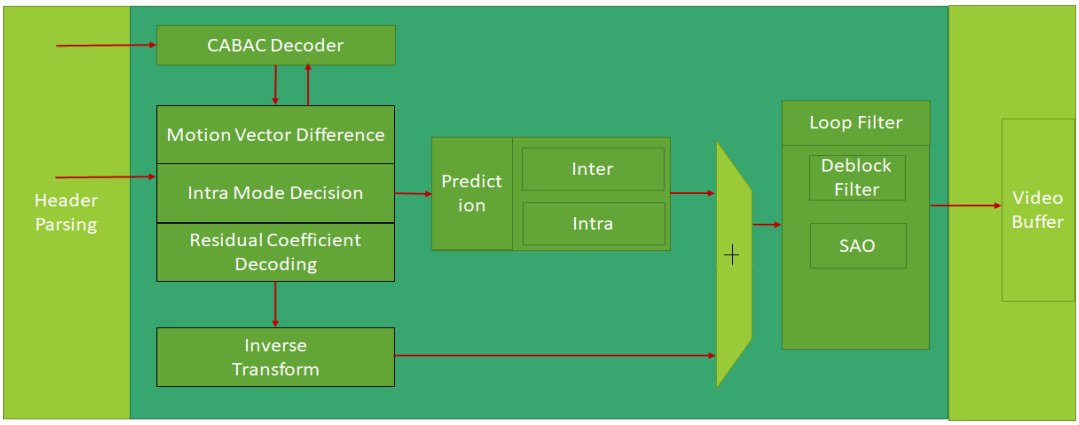


Fig. 4. MV-HEVC SoC Block Diagram

TABLE I. Comparison between custom design and HLS

	LUTs	DFFs	Slice	BRAMs	Freq (MHz)
Vivado HLS	50566	34955	14944	13	200
Verilog[7]	38790	11762	11343	32	150

TABLE II. Used FPGA Resource in Generating Hardware

Module Name	BRAM	DSP	FF	LUT	Latency	Interval
Decode main	45	470	186487	437438		undef
process slice	41	470	179920	421985		undef
Process chroma	8	154	80434	104911		undef
Process luma	0	180	37631	129804		undef
Prediction chroma	0	18	4835	45980	15~18	15~18
Residual block calculation	0	0	15138	33107		undef
Process iter	0	1	10134	19584		undef
Residual block calculation	0	0	6766	18987		undef
Predict intra 16x16	0	24	1385	7321	37	37
scaling	0	51	3796	4910	375	375
intrainfo3	0	0	270	1116	97~145	97~145
u_e	0	0	290	895		undef

D. Chip Design

Figure 5 shows entire MPW design block diagram. There is external memory, delay cell, and i2c to overcome some limitations of the chip. In order to send and receive encoded image data in real time, a high-speed interface, or SerDes, is required by default. However, since many additional designs are required to implement the corresponding IP, a simple

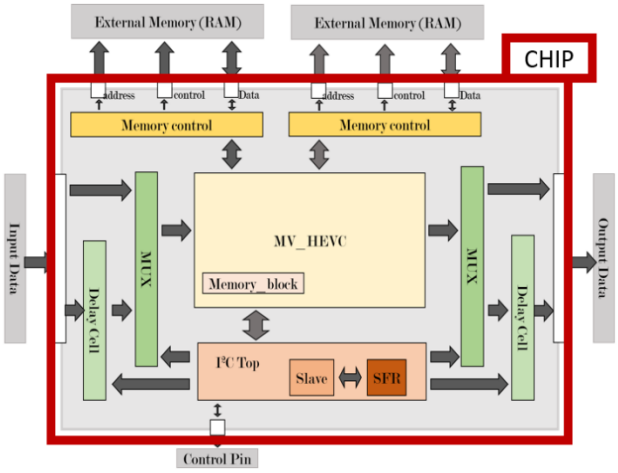


Fig. 5. MPW Design Block Diagram

DDR circuit with DLL was implemented digitally using the GPIO provided in this MPW.

Delay cell was added as shown in Figure 6 to solve the timing violation, which can occur even in the same data input in real chip operation. This can be controlled using a total of 128 layers and adjusted up to 20 ns.

Through the delay cell, timing violation problem can be solved as it shown in Figure 7(a) and (b). Figure 7(a) shows that the signals to be of the same timing are in different states before the delay cell is applied. The difference between the fastest and slowest signals is about 40 ns, which means that if the operating frequency is 25Mhz, it is pushed out by one clock. Therefore, to address this, it is necessary to properly control the 128 steps of the delay cell with I2C to align the same timing as shown in Figure 7(b).

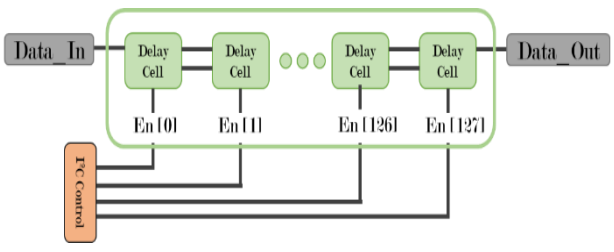
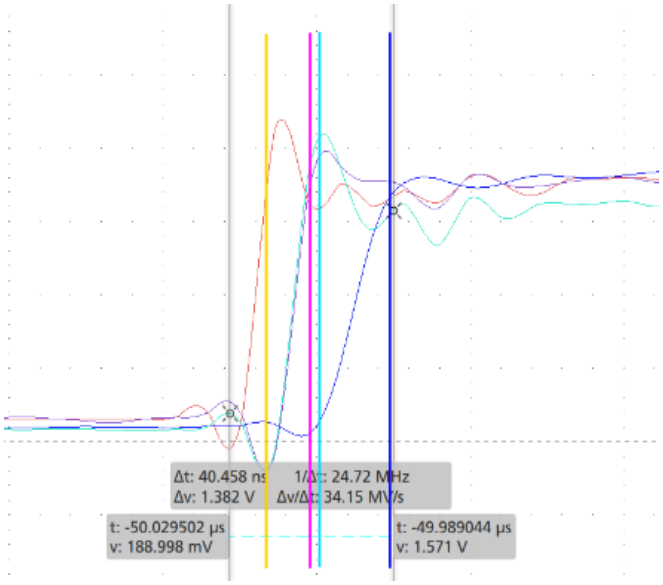
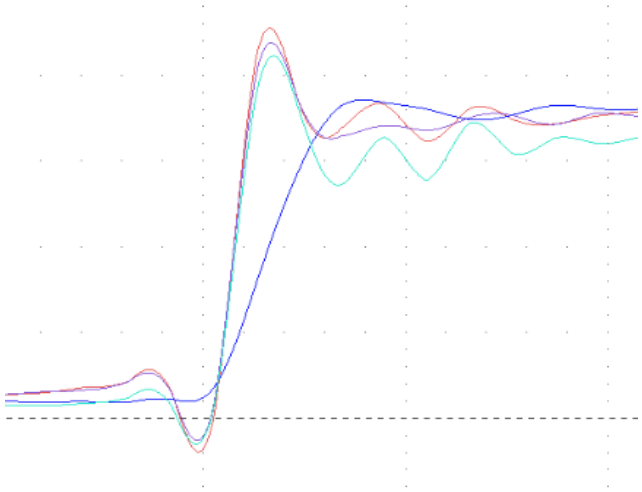


Fig. 6. Delay Cell Block Diagram



(a)



(b)

Fig. 7. (a) Before delay cell (b) After delay cell

In this MPW, additionally, i2c slave and spi slave were designed inside simultaneously to control registers that required initial setting and monitor the condition of the chip according to the situation. MPW can be verified by designing a simple I2c block without occupying a lot of pins in controlling the Delay cell, MUX and External RAM.

In the decoding process, there are many limitations of capacity to perform computations using internal memory. Therefore, external memory was used to solve this problem and the SRAM controller was designed to control it. In orderto solve the I / O pin count problem, it is processed by using FIFO inside the chip. The total number of I / O pins used in the entire system designed using Vivado HLS is 1047. The FIFO was connected to each I / O, and the design was changed to use fewer I / O ports, reducing to 66 as shown in Table III. By using the external ram, it is efficient to control input data interface [8].

TABLE III. (a) Original IO Port (b) Reduction of IO Port

Name	Port	Pin
maul	In	240
	Out	71
PacifiCorp	In	64
	Out	39
pic_memoccupied	Out	12
pic_frame	In	8
	Out	14
⋮	⋮	⋮
sum	In/Out	1024

(a)

Name	Port	Pin
pingpang	In	16
	Out	8
picture	In	6
	Out	3
⋮	⋮	⋮
sum	In/Out	66

(b)

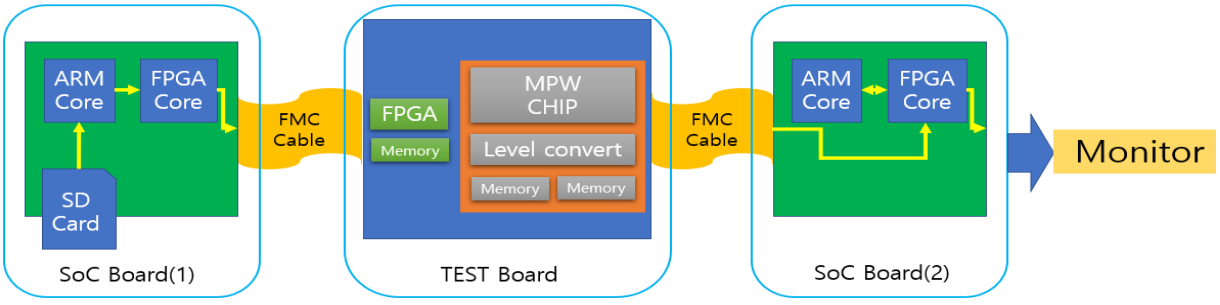


Fig. 8. Test Board Design

III. RESULTS AND DISCUSSIONS

A. Test Environment

Figure 8 shows a block diagram of the chip test environment. A total of two SOC boards and two external memory cards were used, and the soc board used Xilinx company's zed board and the external memory used the memory of Alliance Memory. The test sequence is as follows.

1. Perform the required initial setup between decoding using i2c on the soc board (1).
2. Use FMC cable to send an encoding image stored in the SD card.
3. Process all decoding operations inside the chip & load external memory
4. Output the decoded image data as a chip from the soc board (2), in HDMI.

In this process, due to the operating frequency limit of external memory, the operation was performed at 20 Mhz instead of the initially proposed 100 Mhz, which reduced the data processing speed from what was expected.

B. Result

The memory used in this study can be used in both 8-bit mode and 16-bit mode.

TABLE IV. Operation Power

Unit	Power
Core Power	4.0738 mW
Leakage Power.	. 0.6719 mW

.

Table IV shows the power measurement result. Operation voltage was 1.2v. In particular, to obtain accurate delay value, delay control such as delay cell was made separately and pasted on the whole chip.

IV. CONCLUSION

HEVC is widely used in mobile and other industries. In particular, demand for MV-HEVCs is increasing on mobile platforms. However, MV-HEVC, implemented in standard C++, is not suitable for mobile platforms and needs to be implemented with hardware. so, we implemented it first with the FPGA and then expanded it to the ASIC.

Initial functions required during MV-HEVC checks were controlled using i2c, which resulted in significant pin reduction. It also facilitated control of various blocks of detail, such as delay cell control, MUX, and external setting.

Due to the nature of video processing, many of the memory was used, and many of the constraints were followed, so we tried to solve this problem by using external memory. However, due to limitations in operating frequency of external memory, real-time computational processing with high resolution was somewhat difficult as it fell from the originally intended operating speed. Therefore, the process of amending and supplementing this part is necessary.

ACKNOWLEDGMENT

This work is supported by IDEC for EDA Tool and MPW (Chip fabrication).

REFERENCES

[1] D. A. Johns and K. Martin, *Analog Integrated Circuit Design*, Jonh Wiley & Sons, p. 188, 1997.

[2] S. H. Jung, N. S. Jung, J. T. Hwang, and G. H. Gho, *An integrated CMOS DC-DC converter for battery-operated systems*, Power Electronics Specialists Conference, 1999. PESC 99. 30th Annual IEEE, vol.1, no. 2, pp.43, 47, Aug 1999.

[3] R. W. Erickson and D. Maksimovic, *Fundamentals of Power Electronics*, Springer, 2001.

[4] Cong, J., Liu, B., Neuendorffer, S., Noguera, J., Vissers, K., Zhang, Z., *High-Level Synthesis for FPGAs: From Prototyping to Deployment, Computer-Aided Design of Integrated Circuits and Systems*, IEEE Transactions on, 30(4), 473–491, 2011

[5] Rafal Kulaga and Marek Gorgon, *FPGA Implementation of Decision Trees and Tree Ensembles for Character Recognition in Vivado HLS*, 2015

[6] Jason Cong, Fellow, IEEE, Bin Liu, Stephen Neuendorffer, *High-Level Synthesis for FPGAs: From Prototyping to Deployment*, IEEE, 2011

[7] E. Kalali, E. Ozcan, O. M. Yalcinkaya, I. Hamzaoglu, "A low energy HEVC inverse transform hardware," IEEE Transactions on Consumer Electronics, 2014

[8] K.L. Yeung and S. Hai, *Throughput analysis for input-buffered ATM switches with multiple FIFO queues per input port*, IEEE, 1997

Hye Bin Ha received the B.S. in electrical engineering from Konkuk University, Seoul, Korea, in 2019. She is currently working toword an M.S. at Konkuk University. Her research interests include Embedded system design and FPGA Design for neural network.

Sang Un Park received the B.S. in Electrical Engineering from Shinhan Univ in 2011 and 2018. He is currently working toward an M.S at Konkuk Univ. His research interests include Embedded system design and FPGA Design for neural network device.

Wei Liu received M.S. degree from Konkuk University (2016) and a Bachelor's in Technology degree in Automation from Shenyang Ligong University, Shenyang, China, in 2014. He is currently working toward a Ph.D. degree in Electronics Engineering at Konkuk University in Seoul, South Korea. His research interests include system-on-chip (SoC) design, image processing, multi-core processor systems.

Yong Beom Cho received a B.Sc. degree from Kyongbuk University (1981), a M.Sc. degree from the Univ. of S. Carolina (1988) and a Ph.D. degree from Case Western Reserve University, OH, USA (1992). He is currently a professor in the Department of Electronics Engineering at Konkuk University, Seoul, Korea. His research interests include embedded system design, SoC design, networking systems, application of image processing to mobile environments, and digital communication system design for mobile and ad-hoc network.