International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:7, No:8, 2013

Design of Low-Area HEVC Core Transform Architecture

Seung-Mok Han, Woo-Jin Nam, and Seongsoo Lee

***Abstract*—**This paper proposes and implements an core transform architecture, which is one of the major processes in HEVC video compression standard. The proposed core transform architecture is implemented with only adders and shifters instead of area-consuming multipliers. Shifters in the proposed core transform architecture are implemented in wires and multiplexers, which significantly reduces chip area. Also, it can process from 4×4 to 16×16 blocks with common hardware by reusing processing elements. Designed core transform architecture in 0.13um technology can process a 16×16 block with 2-D transform in 130 cycles, and its gate count is 101,015 gates.

***Keywords*—**HEVC, Core transform, Low area, Shift-and-add, PE reuse.

I. Introduction

RECENTLY, quality and resolution of video images increase continuously. The advent of UHD (ultra-high definition) image requires new video compression techniques beyond H.264/AVC (advanced video coding) [1], since UHD image requires more compression ratio and more computations.

Therefore, a new video compression international standard, HEVC (high-efficiency video coding) [2], has been developed. It was proposed and approved by JTC-VC (joint collaborative team on video coding), a joint research group of ITU-T (international telecommunication union – telecommunication standardization sector) VCEG (video coding experts group) and ISO/IEC (international standard organization/international electro-technical commission) MPEG (moving picture experts group).It significantly improves the compression ratio, efficiently covers very large image sizes such as 2k and 4k UHD, and provides parallel-processing features for easy hardware implementation. Its final version was published on Feb. 2013.

In this paper, a new architecture for HEVC core transform was proposed for high-speed and low-power implementation. It is implemented with only adders and shifters instead of area-consuming multipliers. Its shifters are implemented in wires and multiplexers, which significantly reduces chip area. Also, it can process from 4×4 to 16×16 blocks with common hardware by reusing processing elements.

II. HEVC Core Transform

Fig. 1 shows a block diagram of HEVC encoder. Its major processes are core transform, quantization, intra prediction, motion estimation, motion compensation, sample adaptive offset filter, and context adaptive binary arithmetic coding.

Like H.264/AVC transform, HEVC core transform processes 1-D transforms twice (once horizontal and once vertical) for 2-D transform. HEVC core transform has the following 3 characteristics.

First, in HEVC core transform, forward and inverse transforms exploit same coefficients with transpose [3], as shown in Fig 2. It means that both forward and inverse transforms can be processed by same hardware with transpose.

Second, in HEVC core transform, butterfly architecture can be applied only to only even parts, not to odd part [4], as shown in Fig. 3. This means that the number of multipliers can be reduced in only even parts by exploiting butterfly architecture.

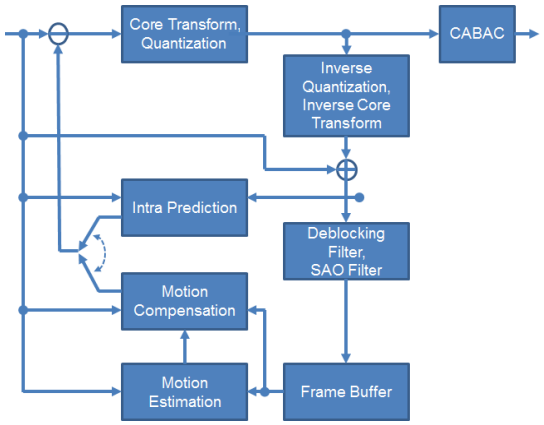

Fig. 1 Block diagram of an HEVC encoder

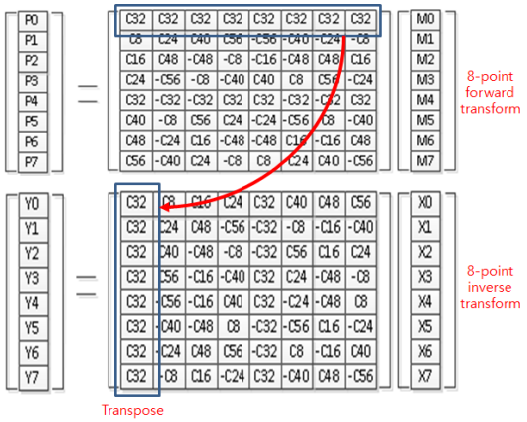

Fig. 2 Relationship between forward and inverse transforms in HEVC core transform

Seong-Mok Han is with Soongsil University, 511 Sangdo-dong, Dongjak-gu, Seoul, 156-743, Korea (email: gkstmdahr0@ssu.ac.kr).
Woo-Jin Nam is with Soongsil University, 511 Sangdo-dong, Dongjak-gu, Seoul, 156-743 Korea (email: nwj@ssu.ac.kr).
Seongsoo Lee is with Soongsil University, 511 Sangdo-dong, Dongjak-gu, Seoul, 156-743 Korea (phone: +82-2-820-0692, email: sslee@ssu.ac.kr).

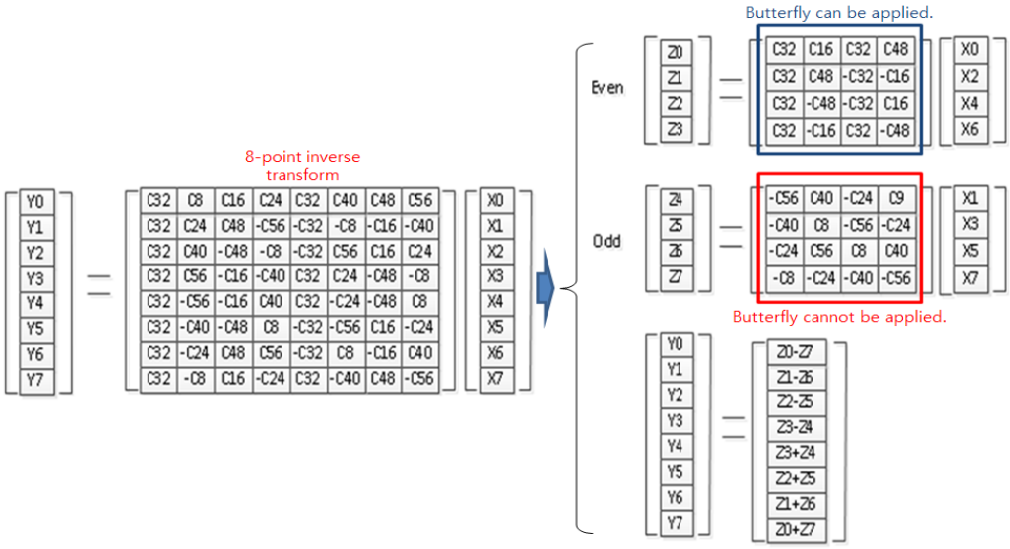

Fig. 3 Butterfly structures of even and odd parts in HEVC core transform

Third, in HEVC core transform, the coefficients of smaller block is same with the coefficients of even part of larger block [5], as shown in Fig. 4. It means that hardware for larger block can also process smaller block by exploiting only even part.

III. HEVC Core Transform

A. Architecture Design

Fig. 5 shows the coefficients of 16-point HEVC core transform. It can be implemented as Fig. 6, and it can process from 4×4 to 16×16 blocks with same hardware [6]. However, it suffers from speed problems in small blocks, since the processing time of 44-point transform are 4 times longer than 1 16-point transform.

In this paper, a new architecture is proposed as shown in Fig. 7. By reusing processing elements, it can concurrently process 4 4-point transforms, or 2 8-point transforms, or 1 16-point transforms. Figs. 7, 8, and 9 show 16-point, 8-point, and 4-point transforms, respectively.

B. Processing Element Design

Coefficients of HEVC core transform are fixed constants, so it can be implemented by adders and shifters without multipliers [3], [7]. Considering the coefficients in Fig. 5, they have 28 different absolute values as shown in Table I. It shows that all coefficients can be calculated by maximum 3 additions/subtractions and maximum 4 shifts. For examples, A*75 = A*(64+8+2+1) = (A<<6)+(A<<3)+(A<<1)+(A<<0), and it requires 3 additions and 4 shifts. Similarly, A*87 = A*(64+32-8-1) = (A<<6)+(A<<5)-(A<<3)-(A<<0), and it requires 1 addition, 2 subtractions, and 4 shifts.

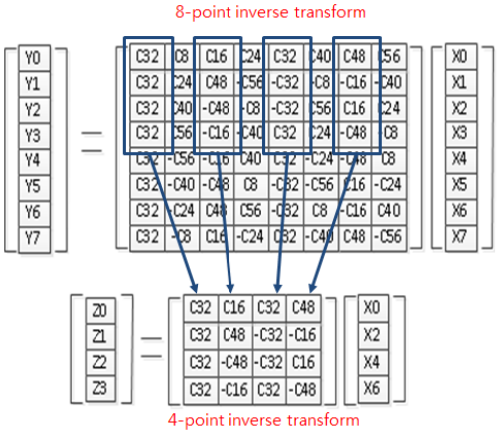

Fig. 4 Relationship between larger size and smaller size blocks in HEVC core transform

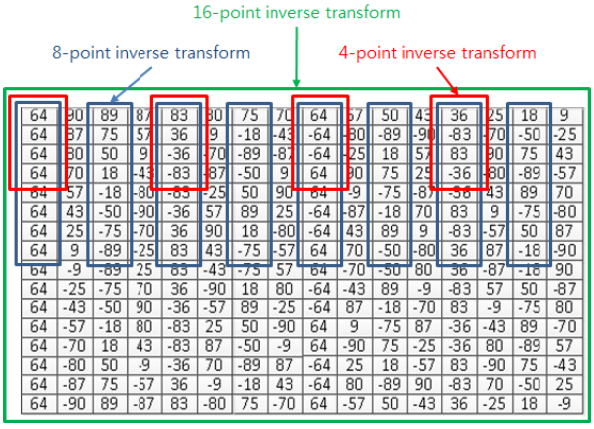

Fig. 5 Coefficients of 16-point HEVC core transform

TABLE I
ADDITION-SHIFT OPERATIONS OF 16-POINT HEVC CORE TRANSFORM

Coefficients	Addition/Shift Operations		# ofAdditions	# of Subtractions	# ofShift
A*4	=A*(4)	=(A<<2)	0	0	1
A*9	=A*(8+1)	=(A<<3)+(A<<0)	1	0	2
A*13	=A*(8+4+1)	=(A<<3)+(A<<2)+(A<<0)	2	0	3
A*18	=A*(16+2)	=(A<<4)+(A<<1)	1	0	2
A*22	=A*(16+4+2)	=(A<<4)+(A<<2)+(A<<1)	2	0	3
A*25	=A*(16+8+1)	=(A<<4)+(A<<3)+(A<<0)	2	0	3
A*31	=A*(32-1)	=(A<<5)-(A<<0)	0	1	2
A*36	=A*(32+4)	=(A<<5)+(A<<2)	1	0	2
A*38	=A*(32+4+2)	=(A<<5)+(A<<2)+(A<<1)	2	0	3
A*43	=A*(32+8+2+1)	=(A<<5)+(A<<3)+(A<<1)+(A<<0)	3	0	4
A*46	=A*(32+8+4+2)	=(A<<5)+(A<<3)+(A<<2)+(A<<1)	3	0	4
A*50	=A*(32+16+2)	=(A<<5)+(A<<4)+(A<<1)	2	0	3
A*54	=A*(32+16+4+2)	=(A<<5)+(A<<4)+(A<<2)+(A<<1)	3	0	4
A*57	=A*(32+16+8+1)	=(A<<5)+(A<<4)+(A<<3)+(A<<1)	3	0	4
A*61	=A*(64-2-1)	=(A<<6)-(A<<1)-(A<<0)	0	2	3
A*64	=A*(64)	=(A<<6)	0	0	1
A*67	=A*(64+2+1)	=(A<<6)+(A<<1)+(A<<0)	2	0	3
A*70	=A*(64+4+2)	=(A<<6)+(A<<2)+(A<<1)	2	0	3
A*73	=A*(64+8+1)	=(A<<6)+(A<<3)+(A<<0)	2	0	3
A*75	=A*(64+8+2+1)	=(A<<6)+(A<<3)+(A<<1)+(A<<0)	3	0	4
A*78	=A*(64+8+4+2)	=(A<<6)+(A<<3)+(A<<2)+(A<<1)	3	0	4
A*80	=A*(64+16)	=(A<<6)+(A<<4)	1	0	2
A*83	=A*(64+16+2+1)	=(A<<6)+(A<<4)+(A<<1)+(A<<0)	3	0	4
A*85	=A*(64+16+4+1)	=(A<<6)+(A<<4)+(A<<2)+(A<<0)	3	0	4
A*87	=A*(64+32-8-1)	=(A<<6)+(A<<5)-(A<<3)-(A<<0)	1	2	4
A*88	=A*(64+16+8)	=(A<<6)+(A<<4)+(A<<3)	2	0	3
A*89	=A*(64+16+8+1)	=(A<<6)+(A<<4)+(A<<3)+(A<<0)	3	0	4
A*90	=A*(64+16+8+2)	=(A<<6)+(A<<4)+(A<<3)+(A<<1)	3	0	4

Fig. 10 shows the architecture of the proposed processing element to process the dataflow of Fig. 7. Shifters in Fig. 10 performs shift operations from 0 to 6 bits, but it is implemented by only hardwire connections without additional circuits. Therefore, the processing element is implemented by only multiplexors, AND gates, adders, and subtractors.

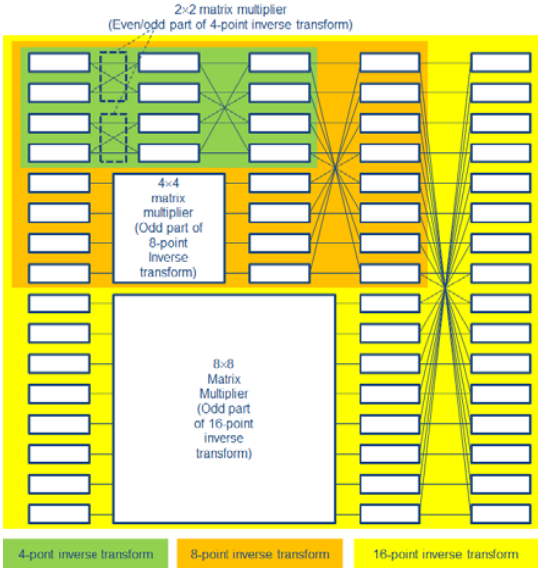

Fig. 6 Conventional HEVC core transform architecture

In Fig. 10, two inputs of a processing element are selected among the outputs of other processing elements. There are 26 processing elements in Fig. 7, so 26:1 MUX is required. However, by careful input assignment, first and second inputs of a processing element can be selected among maximum 3 and 5 outputs of other processing elements, respectively. Therefore, 3:1 and 5:1 MUXs are used in Fig. 10.

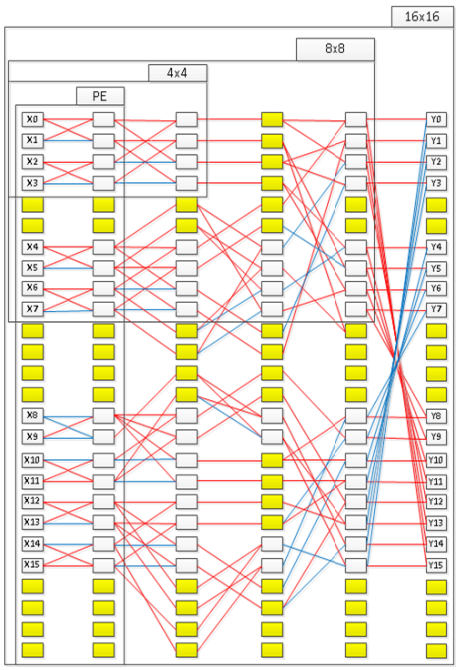

Fig. 7 Proposed architecture of HEVC core transform

C. Gate Counts

In this paper, the conventional and the proposed architectures are designed and synthesized in 0.13um technology to compare gate counts. Proposed architecture exploits 1 units of core transform block in Fig. 7, while conventional architecture exploits 4 units of core transform blocks in Fig. 6 to have same throughput with proposed architecture.

From the synthesis results, gate counts of the proposed and conventional architectures are 101,013 and 292,325 gates, respectively. Therefore, the proposed architecture reduces the gate counts to about 1/3 with same throughput.

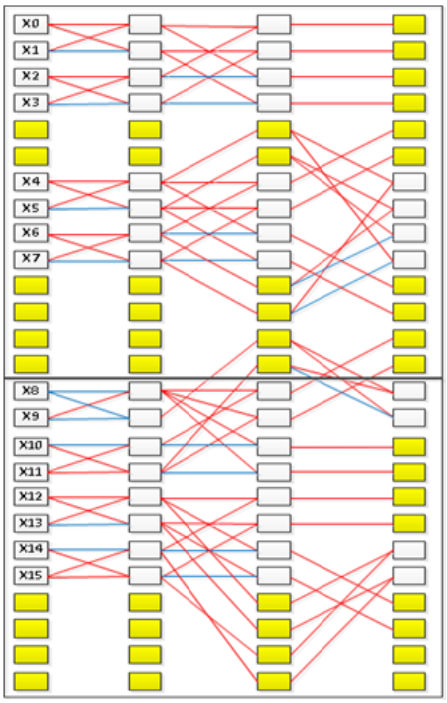

Fig. 8 8-point inverse transform operation in the proposed architecture

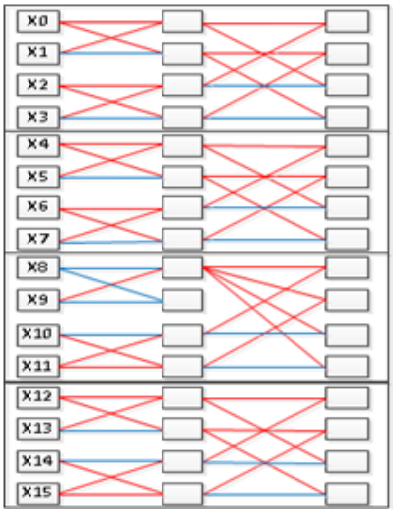

Fig. 9 4-point inverse transform operation in the proposed architecture

IV. Conclusions

This paper proposes and implements a low-areaHEVC core transform architecture. It is implemented with only adders and shifters instead of area-consuming multipliers. Shifters in the proposed core transform architecture are implemented in wires and multiplexers, which significantly reduces chip area. Also, it can process from 4×4 to 16×16 blocks with common hardware by reusing processing elements. Designed core transform architecture in 0.13um technology can process a 16×16 block with 2-D transform in 130 cycles, and its gate count is 101,015 gates.

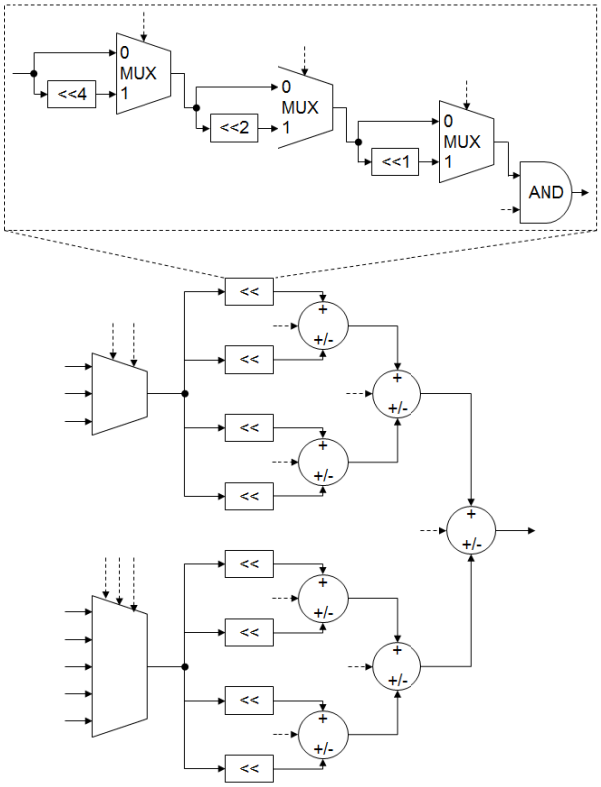

Fig. 10 Proposed processing unit of HEVC core transform

Acknowledgment

This research was supported by the MSIP (Ministry of Science, ICT&Future Planning), Korea, under University ITRC support program (NIPA-2013-H0301-12-2006) supervised by the NIPA(National IT Industry Promotion Agency).

This research was also supported by ETRI SW-SoC R&BD Center, Human Resource Development Project, funded by the MOTIE (Ministry of Trade, Industry, & Energy), Korea (10039239).

References

[1] T. Wiegand, G. Sullivan, G. Bjontgaard, and A. Luthra, "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, Jul. 2003.

[2] G. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", *IEEE Transactions on*

Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.
[3] A. Fuldseth, G. Bjøntegaard, and M. Budagavi, "CE10: Core Transform Design for HEVC," *JCTVC-G495*, Nov. 2011.
[4] J. Park, W. Nam, S. Han, and S. Lee, "High Efficiency Video Coding(HEVC) 16x16 & 32x32 Inverse Transform IP Design for Large-Scale Displays", *Proceedings of International Technical Conference on Circuits/Systems, Computers, and Communications*, pp. 153-155, Jun. 2011.
[5] M. Budagavi, V. Sze, and M. Sadafale, "Hardware analysis of transform and quantization," *JCTVC-G132*, Nov. 2011.
[6] M. Budagavi and V. Sze, "Unified Forward+Inverse Transform Architecture for HEVC", *Proceedings of IEEE International Conference on Image Processing*, pp. 209-212, 2012.
[7] J. Park, W. Nam, S. Han, and S. Lee, "2-D Large Inverse Transform (16x16, 32x32) for HEVC (High Efficiency Video Coding)", *Journal of Semiconductor Technology and Science*, vol. 12, no. 2, pp. 203-211, Jun. 2012.

Seung-Mok Han received B.S. degree in E.E from Soongsil University, Korea, in 2011. He is currently pursuing M.S degree in E.E. Soongsil University, Korea. His research interests include high efficiency video coding, semiconductor, digital circuit, multimedia, and ultra-low-power analog circuit.

Woo-Jin Nam received B.S. degree in E.E from Soongsil University, Korea, in2011. He is currently pursuing M.S degree in E.E. Soongsil University, Korea. His research interests include high efficiency video coding, battery life time enhancement technology, and low-power system architecture.

Seongsoo Lee received B.S, M.S, and Ph.D. degrees in E.E. from Seoul National University, Korea in 1991, 1993, and 1998, respectively. In1998-2000, he was a research associate in Institute of Industrial Science, University of Tokyo, Japan. In 2000-2002, he was a research professor in Department of Electronic Engineering, Ewha Womans University, Korea. He joined School of Electronic Engineering at Soongsil University, Korea in 2002, where he is currently an associate professor. His research interests include low-power SoC, multimedia SoC, and battery management SoC.