An 8K H.265/HEVC Video Decoder Chip With a New System Pipeline Design

Dajiang Zhou, *Member, IEEE*, Shihao Wang, Heming Sun, Jianbin Zhou, Jiayi Zhu, Yijin Zhao, Jinjia Zhou, *Member, IEEE*, Shuping Zhang, Shinji Kimura, *Member, IEEE*, Takeshi Yoshimura, *Member, IEEE*, and Satoshi Goto, *Life Fellow, IEEE*

***Abstract*— 8K ultra-HD is being promoted as the next-generation video specification. While the High Efficiency Video Coding (HEVC) standard greatly enhances the feasibility of 8K with a doubled compression ratio, its implementation is a challenge, owing to ultrahigh-throughput requirements and increased complexity per pixel. The latter comes from the new features of HEVC. At the system level, the most challenging of them is the enlarged and highly variable-size coding/prediction/transform units, which significantly increase the requirement for on-chip memory as pipeline buffers and the difficulty in maintaining pipeline utilization. This paper presents an HEVC decoder chip featuring a system pipeline that works at a nonunified and variable granularity. The pipeline saves on-chip memory with a novel block-in-block-out queue system and a parameter delivery network, while allowing overhead-free and fully pipelined operation of the processing components. With the system pipeline design combined with various component-level optimizations, the proposed decoder in 40 nm achieves a maximum throughput of 4 Gpixels/s or 8K 120 frames/s for the low-delay-P configuration of HEVC, 7.5–55 times faster than prior works. It supports 8K 60 frames/s for the low-delay and random-access configurations. In a normalized comparison, it also shows 3.1–3.6 times better area efficiency and 31%–55% superior energy efficiency.**

***Index Terms*— 4K, 8K, ASIC, block-in-block-out (BIBO) queue, H.265/High Efficiency Video Coding (HEVC), Super Hi-Vision, UHDTV, ultra-HD, video decoder.**

I. Introduction

8K ULTRA-HD, also known as Super Hi-Vision, is being promoted as the next-generation digital video specification. By delivering up to 10 b/sample, 7680 × 4320 pixels/frame, and 120 frames/s, 8K achieves remarkably improved visual experience. To store and transmit the huge volume of 8K video data, highly efficient and real-time compression and decompression are essential. From a communication channel perspective, the latest High Efficiency Video Coding standard (H.265/HEVC) [1] has greatly enhanced the feasibility of 8K by doubling the compression ratio relative to its predecessor H.264/AVC [2]. The deeper compression, however, comes at the cost of a higher complexity per pixel at the source video codec. Compared with a mainstream 1080p 30 frames/s H.264 decoder, the next-generation 8K ultra-HD decoder for HEVC involves 110–160 times complexity, with 80 times from the higher data throughput and the rest 1.4–2 times [3] from the advanced features of HEVC.

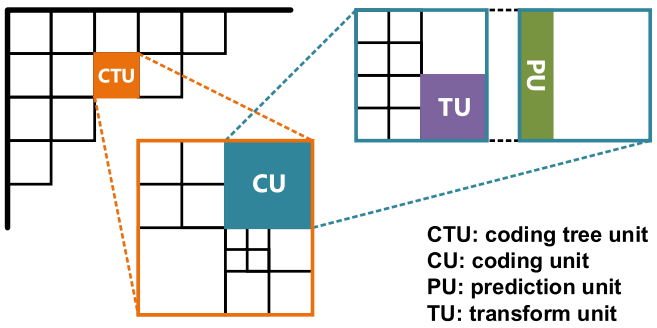


Fig. 1. Data partitioning in H.265/HEVC and similar video coding frameworks.

In HEVC, many new or upgraded coding tools are applied. The highly hierarchical coding tree units (CTU) that can vary from 16 × 16 up to 64 × 64 replaced the fixed-size 16 × 16 macroblocks (MBs) in H.264. A CTU can be further broken down into coding units (CUs), prediction units (PUs), and transform units (TUs) of various sizes, as shown in Fig. 1. At the component level, the extension of maximum transform size to 32 × 32 significantly increases computational complexity. In intra prediction, up to 35 prediction modes, including DC, planar, and 33 angular modes, are supported. In inter prediction, an eight-tap or seven-tap interpolation filter (relative to the six-tap version in H.264), asymmetric motion partition, and advanced motion vector (MV) prediction are introduced. In spite of the simplification of the deblocking filter (DBF), a new in-loop filter named sample adaptive offset (SAO) is introduced. In entropy coding, context adaptive binary arithmetic coding (CABAC) is adopted.

To achieve the high throughput required by 8K HEVC decoding with an efficient VLSI design, the main challenges are in: 1) external memory bandwidth requirements; 2) complexity and data dependencies at the component level; and 3) pipeline integration at the system level. For 1), since HEVC decoding shares very similar memory access patterns to H.264 decoding, many existing techniques developed for the latter can be applied. Several architectures for optimizing

Manuscript received May 1, 2016; revised July 31, 2016 and September 19, 2016; accepted September 28, 2016. Date of publication November 4, 2016; date of current version January 4, 2017. This paper was approved by Guest Editor Dejan Markovic. This work was supported in part by the Regional Innovation Strategy Support Program of Ministry of Education, Culture, Sports, Science and Technology, Japan, and in part by NEC Corporation.

The authors are with the Graduate School of Information, Production and Systems, Waseda University, Kitakyushu 808-0135, Japan (e-mail: zhou@fuji.waseda.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSSC.2016.2616362

the memory traffic of the HEVC decoders have also been discussed in [4]–[7]. For 2), the component-level issues for the VLSI design of HEVC decoders have been studied in many previous works, including contributions to CABAC decoding [8]–[11], parameter decoding [12], motion compensation [7], [13]–[17], inverse transforms (ITs) [18]–[23], intra prediction [24]–[28], and in-loop filters [29]–[33]. Among these designs, [7], [12], [27], and [29] achieved the throughput of 16 pixels/cycle or above, which is required for real-time 8K decoding at a reasonable clock rate.

For 3), at the system level, the enlarged and highly variable-size CTU/CU/PU/TU remains a key challenge to the design of an efficient pipeline that integrates the decoder's components. The conventional CTU-level (MB-level) pipelining will lead to huge requirements for on-chip memory as pipeline buffers. On the other hand, though it is possible to pipeline the decoder at a smaller unified granularity (e.g., 16×16) to reduce the buffer space, doing so may notably decrease pipeline utilization, as analyzed in [34]. This is expensive in terms of both chip area and energy consumption, particularly for an 8K-level HEVC decoder that has a very limited clock cycle budget for each CTU. System-level design of HEVC decoders has been presented in [34]–[37] as ASIC chips or designs, and in [38] and [39] as FPGA-based implementations, with their specifications ranging from 1080p to 4K. So far, no previous HEVC decoder chips have achieved 8K or Gpixel/s-level performance.

This paper presents the system-level design of an H.265/HEVC decoder for 8K ultra-HD applications. It features a system pipeline that works at a nonunified and variable granularity. Based on a novel block-in-block-out (BIBO) queue building block and a parameter delivery network, the proposed pipeline allows the processing components to be synchronized locally using reduced memory space as pipeline buffers. It also allows overhead-free and fully pipelined operation of the processing components, which results in a high pipeline utilization. The proposed design was implemented in a test chip, which delivers a maximum throughput of 4 Gpixels/s (8K 120 frames/s) and 2 Gpixels/s (8K 60 frames/s), respectively, for unidirectional and bidirectional interpredicted HEVC decoding.

The rest of this paper is organized as follows. Section II gives an overview of the top-level architecture of the chip. Section III presents the functionality and architecture of the BIBO queue building block. Section IV describes the design philosophy of the system pipeline. Section V shows details on how the processing components are integrated into the pipeline. Sections VI and VII give the implementation results and the conclusion, respectively.

II. Architecture Overview

Fig. 2 shows the top-level system architecture. The chip comprises two primary domains for video decoder functionality and external DRAM connectivity, respectively, both clocked by on-chip phase-locked loops (PLLs). The decoder domain working at 0–300 MHz consists of processing components starting from the high-level parameter parser (HPP) and ending with the frame writer (FrmWr). Taking in the input bit stream, HPP parses the parameters at the sequence, picture, and slice header levels. The remaining low-level parameters, including residuals, prediction modes, and various flags, are decoded by the CABAC decoder (CABAD) and synthesized [for MVs and boundary strengths (BSs)] by the parameter decoder (PDec). The parameters are then distributed to the pixel processing components, including the inverse quantizer (IQ), ITs (IT1 and IT2), intrapredictor (Intra), frame reader (FrmRd), motion compensator (MC), reconstructor (Rec), DBF, and SAO filter. Finally, FrmWr writes the decoded pixels from SAO into the DRAM-based frame buffer. PDec, FrmRd, and FrmWr share the DRAM bandwidth through a 256-b bus, which is bridged to the DRAM domain through asynchronous buffers. Compared with other choices, such as 128 and 512, the 256-b bus achieves a balance between the required operating frequency (lower than the 128-b bus) and buffer cost. The DRAM domain consists of the DDR3 controller and peripheral (PHY) that follows a fixed 1:2 clock ratio. With the controller working at up to 400 MHz, a maximum data rate of DDR3-1600 can be supported at the 64-b DRAM interface. The clock rates of the decoder and DRAM domains are independently tunable to enable a flexible combination of computational capability and DRAM bandwidth for various application scenarios. An externally clocked host interface connects the primary domains to a host FPGA, allowing the latter to feed test configurations and bit streams into, and to retrieve decoded pixels and debugging data from the chip.

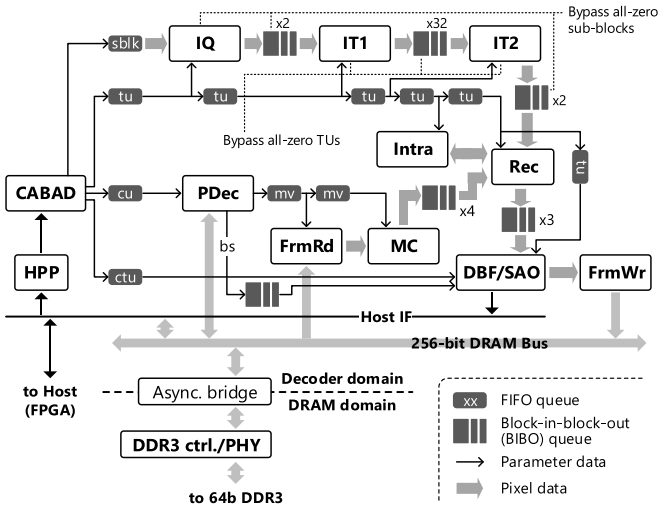


Fig. 2. Chip block diagram.

III. BIBO Queue

A first-in-first-out (FIFO) queue is among the simplest multiword data storage structures, which is therefore the most preferable to be used as intercomponent communication buffer. Video decoding, however, involves various data reordering operations between its components, making random addressability a must. The BIBO queue is therefore designed to combine the features of an FIFO queue and a randomly addressable array.

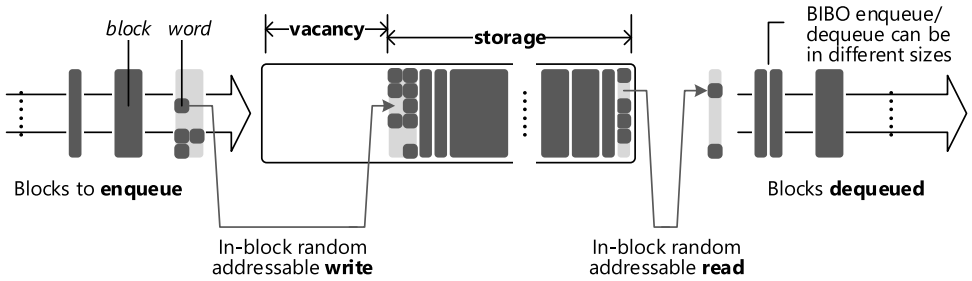


Fig. 3. BIBO queue.

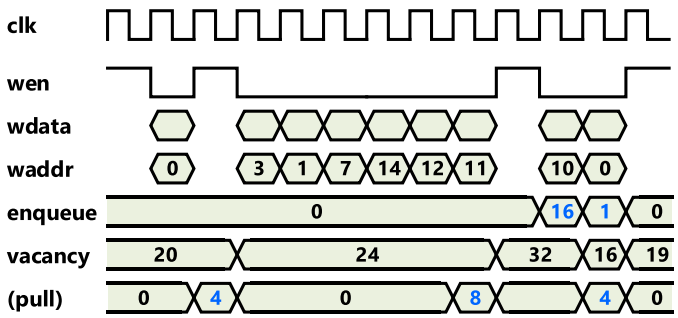


Fig. 4. Write and enqueue timing of a BIBO queue. In this example, two blocks (sized 16 and 1, respectively) were enqueued into the BIBO. For the first block, only a subset of words (8 out of 16) were explicitly written, while all the rest words were treated as zeros by the zero-map function of BIBO.

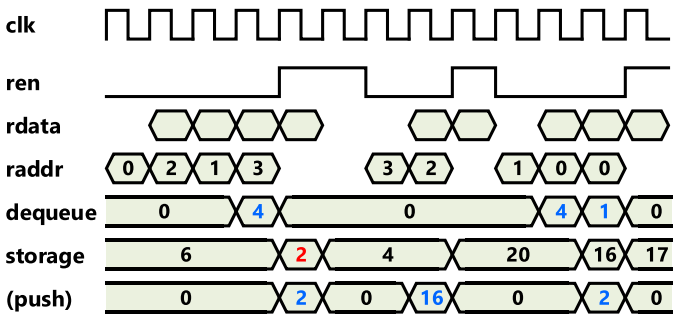


Fig. 5. Read and dequeue timing of a BIBO queue. In this example, three blocks (sized 4, 4, 1) were dequeued from the BIBO. For the second block, read operations could not be conducted until the storage level was no lower than the block size (4).

Similar to an FIFO queue, the BIBO queue has two interfaces dedicated to write and read, respectively, as shown in Fig. 3. In addition, it has a two-level data structure of words and blocks. A word is a basic access unit that can be written into or read from the BIBO in a single clock cycle. A block comprises multiple words. At the word level, write and read accesses are allowed to be in a randomly addressable order inside each block. When all the write/read accesses to the current block are finished, an enqueue/dequeue operation should be performed for the block to indicate its completion. At the block level, enqueue/dequeue operations must follow an FIFO manner. Blocks can be variable in size. They can also be freely combined or separated between the write and read ports of the BIBO, i.e., a single block enqueued into the BIBO can later be dequeued multiple times as multiple smaller blocks, and vice versa. Fig. 4 shows the write and enqueue timing. Writing into a BIBO is similar to writing into an RAM except that the address to specify (via port *waddr*) is a relative in-block address, which is therefore always smaller than the block size. A nonzero value at port *enqueue* specifies an enqueue operation as well as the size of the block enqueued, which is most efficient if given at the same clock cycle for writing the last word of the current block. Port *vacancy* reflects the current level of usable space, according to which it can be gauged at the write interface whether a block can be written or must be stalled by comparing the block size to *vacancy*. Fig. 5 shows the read and dequeue timing. Similarly, *raddr* is a relative in-block address. A nonzero value at port *dequeue* specifies the dequeue operation and the block size, while *storage* reflects the number of words available in the BIBO.

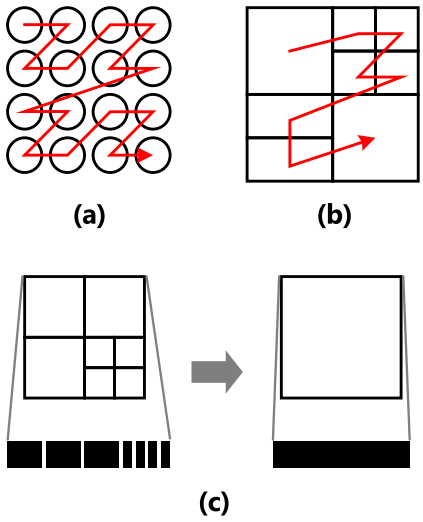


Fig. 6. (a) Z-scan of pixels. (b) Z-scan of blocks. (c) Block merging in BIBO. For all types of blocks, vertical nonrectangular splitting is allowed, since it does not conflict with pixel-level Z-scan.

The block combination and separation feature can be especially useful in an HEVC decoder, since various processing components of the decoder tend to operate in different granularities. In the proposed system pipeline, a block is usually mapped to a CTU/CU/PU/TU depending on the processing granularity, while a word is mapped to a fixed number (e.g., 4×4) of pixels. As long as a Z-scan order is followed at both the pixel and the block levels, as shown in Fig. 6,

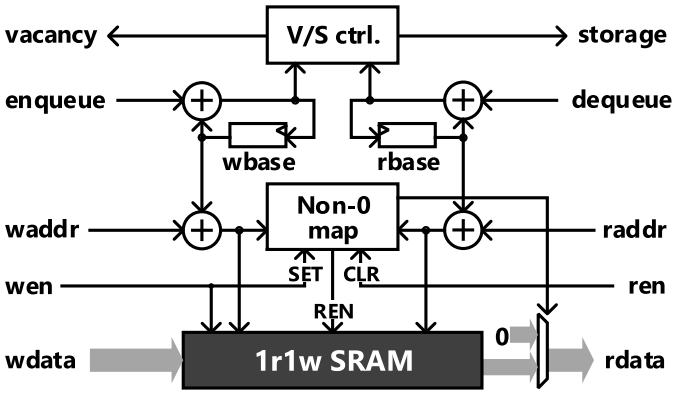


Fig. 7. Implementation of the BIBO queue.

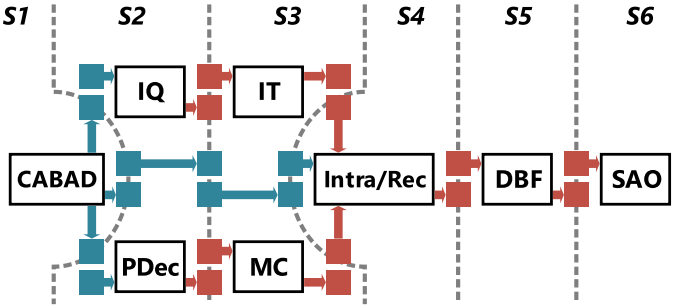


Fig. 8. Example of conventional CTU-level pipelining. The decoder is divided into six pipeline stages interfaced with multiple buffering.

TABLE I
COMPARISON OF CTU/CU/PU/TU SIZES. MB AND SUB-MB IN H.264 ARE REGARDED AS CORRESPONDING TO CTU AND CU IN HEVC

	H.264/AVC	H.265/HEVC	Increase of space requirements
CTU	16×16	16×16~64×64	16× pixels
CU	8×8~16×16	8×8~64×64	16× parameters
PU	4×4~16×16	4×4~64×64 (intra) 8×4/4×8~64×64 (inter)	8~16× parameters
TU	4×4~8×8	4×4~32×32	16× parameters

combination and separation of 2-D CTU/CU/PU/TUs are equivalent to those of 1-D blocks, which can therefore be automated by the BIBO. For a processing component, pixel-level (or word-level) Z-scan can be supported with a look-up table-based address translation regardless of the real order the words are written, thanks to the in-block random addressability. At the block level, the CTU/CU/PU/TUs should be processing following the Z-scan.

Fig. 7 shows the brief concept of BIBO's implementation. A BIBO queue encloses a 1R1W SRAM for storing the words. A base address is maintained for both the write and read interfaces. For a write/read operation, the physical address is generated as the sum of *waddr*/*raddr* (the relative in-block address) and *wbase*/*rbase* (the base address). Upon an enqueue/dequeue operation, the base address is self-incremented by the size of the block enqueued/dequeued. For power savings, the BIBO can optionally maintain a nonzero map to skip the write and read of all-zero words. The nonzero map consists of an array of register bits, each of which corresponds to a word in the SRAM. A bit is set or cleared when its corresponding word is written or read. As a result, only nonzero words need to be explicitly written into the BIBO, with an example shown in Fig. 4. All the remaining words, with their corresponding register bits marked as zero, will be regarded as all-zero at the read interface, regardless of the physical content stored in the SRAM. This mechanism saves the SRAM access power for both write and read, or even allows the upstream processing component to completely skip the computation for a word that has been predicted to generate an all-zero output.

IV. SYSTEM PIPELINE

H.264/AVC decoders are usually pipelined at the MB level [40]–[42]. Fig. 8 shows an example assuming that the same pipelining strategy is directly applied to an HEVC decoder. The decoder's processing components are grouped into several stages according to the data flow and synchronized by a global scheduler. To ensure smooth pipeline operation, all stages follow a common granularity (MB or CTU) while on-chip multiple buffering for both pixels and parameters should be inserted between the stages. Though a double-granularity-sized memory space is a minimum requirement for the multiple buffering, a triple-granularity-sized or even larger space may be desirable to conceal the speed variation of variable-throughput stages (e.g., CABAD and MC) or to address delay of the subpipelines inside the processing components. While such a design style works well for H.264 decoders, the required pipeline buffer space for HEVC increases significantly to approximately 200 kB with the 16 times increase of both size and variability of CTU/CU/PU/TUs, as summarized in Table I.

In this paper, we address the memory space issue while maintaining high pipeline utilization, with a novel system pipeline design. As shown in Fig. 2, parameters from CABAD are classified into four levels: CTU, CU, TU, and subblock (4×4), and distributed through a network of FIFO queues to the target processing components. From PDec, MVs are also delivered by FIFO queues to FrmRd and MC at the PU level, while a BIBO queue is used to deliver BSs to DBF. All pixel buffers between the pixel processing components are implemented using BIBO queues. The pipeline, with an example working flow provided in Fig. 9, has the following features.

A. Nonunified Processing Granularity Through Local Synchronization

Compared with a conventional global synchronization, the proposed pipeline is synchronized locally with each processing component independently tracking the vacancy/storage status of the interfaced FIFO and BIBO queues. This allows the processing components to work in different granularities, as shown in Table II. Having IQ, IT1/IT2, and PDec/MC working at the 4×4 TU and PU levels is also the most straightforward for these components' algorithms, which enables efficient and simplified implementation.

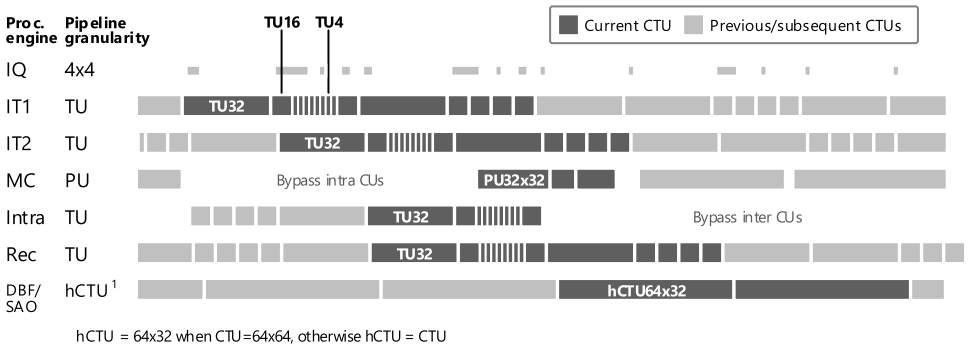


Fig. 9. Example working flow of the proposed system pipeline.

TABLE II
PIPELINE GRANULARITY OF PROCESSING COMPONENTS

	Granularity	Max. granularity in pixels
IQ	4×4	4×4
IT1/IT2	TU	32×32
PDec	PU	-
MC	CU	64×64
Rec	TU	32×32
DBF/SAO	hCTU*	64×32

** hCTU = 64x32 when CTU = 64x64, otherwise hCTU = CTU.

B. Variable Processing Granularity of a Single Component

Since the CTU/CU/PU/TU size varies, as shown in Table I, it is naturally required that the processing granularity can change dynamically. This can be supported by the BIBO's capability to allow pulling/pushing of blocks in a variable size. Moreover, the block combination and separation capability of BIBO address the different sizes in which blocks are enqueued and dequeued at the two ends of the BIBO.

C. Standardized and Overhead-Free Interface Protocol

The processing components are designed to handshake only with FIFO and BIBO queues, or interfaced through a simple FIFO-like valid/ready interface (FrmRd-MC, Intra-Rec, and DBF/SAO-FrmWr). This eases development by avoiding the necessity to define direct handshake timing between the components. Fig. 10 shows a template operation pattern followed by most components of the proposed decoder. The processing of a block (CU/PU/TU) starts as soon as: 1) the input parameter word for this block is available in the upstream FIFO (*ififo*); 2) all the input pixel data for this block are available in the upstream BIBO (*ibibo*); and 3) the downstream BIBO (*obibo*) has enough free space for all the pixel data to be output. The block will be processed in a fully pipelined style and the outputs will be produced after a fixed delay from the subpipeline of the processing component. The input block will be dequeued in the same clock cycle for loading the last word of the block. At the same time, the valid/ready handshake will be completed for the parameter FIFO. Note a single word of the FIFO contains all the parameters for the current block,

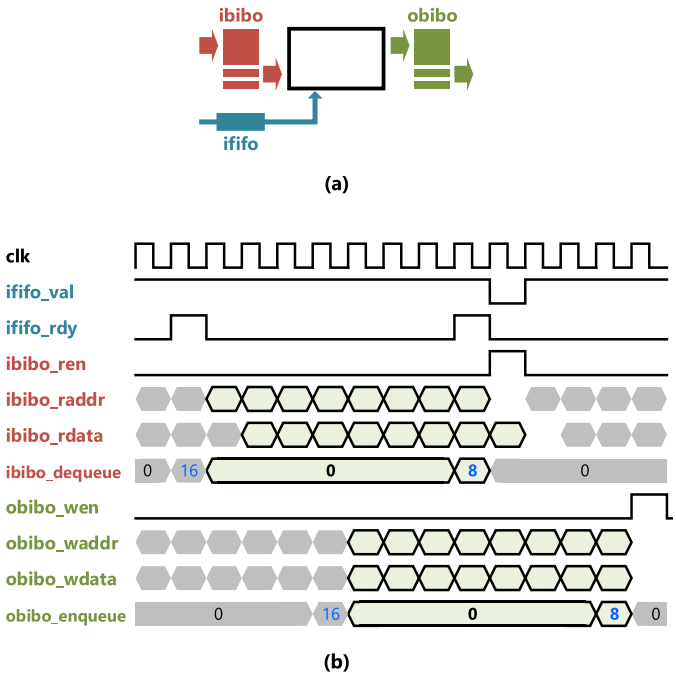


Fig. 10. (a) I/O of a typical processing component. (b) Standard operation pattern of the processing component. Between the input FIFO queue *ififo* and the processing component, a standard valid/ready handshake is followed. The current (focused) block has a size of eight words. The pipeline delay is four clock cycles.

unlike the pixel data usually distributed in multiple words of the BIBO. Once the current block is finished, processing for the next one starts immediately given all availability/vacancy conditions with the upstream/downstream queues are fulfilled, so the handshake does not produce any timing overhead.

D. Reduction of Pixel Buffer Space

Compared with CTU-level multiple buffering, pipeline buffers implemented in BIBO queues can be significantly smaller, since the BIBOs can be sized according to the maximum granularity (enqueue or dequeue size) of its upstream and downstream processing components, rather than CTUs. For smooth operation of the pipeline, the space of each BIBO is assigned to be three times of the maximum granularity, as shown in Table III, out of which two times ensure the upstream and downstream components can always have independent work spaces in the BIBO. The remaining one time of space

TABLE III
SIZING OF BIBO QUEUES

BIBO queue	Enqueue size	Dequeue size	Bits/pixel	Max. enqueue/dequeue	BIBO size	# of Banks
IQ-IT1	4x4	**TU**	16	3KB	9KB	2
IT1-IT2	**TU**	**TU**	16	3KB	9KB	32
IT2-Rec	**TU**	**TU**	11	2.0625KB	11KB	2
MC-Rec	**CU**	TU	10	7.5KB	22.5KB	4
Rec-DBF	4x4	**hCTU**	10	3.75KB	11.25KB	2

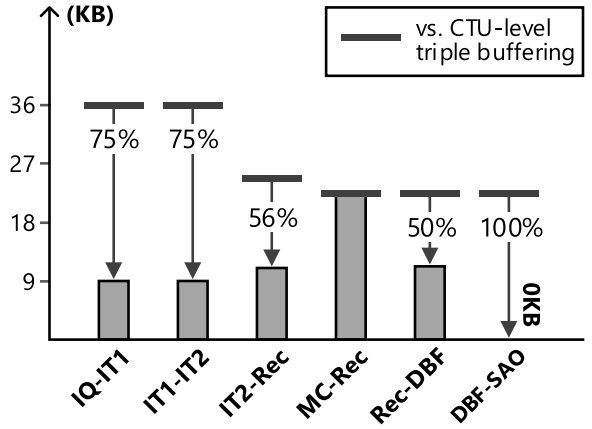


Fig. 11. Reduction of pixel buffer memory space.

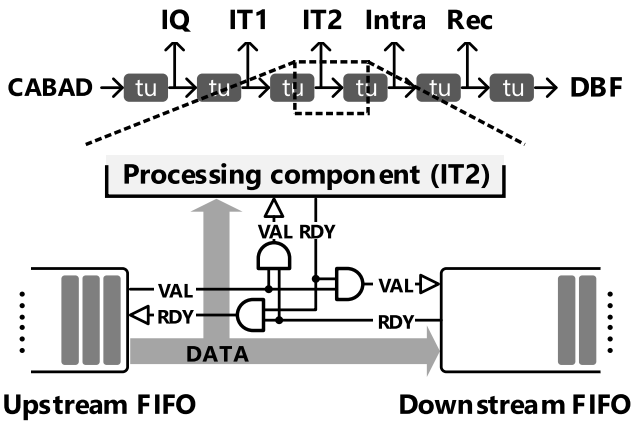


Fig. 12. Relay of parameter FIFO queues.

is preserved to overcome the clock cycle overhead from the subpipeline delay in each component (see Fig. 10) and to alleviate the influence of throughput variation. It should be noted that the former is particularly critical for the targeted 8K performance for which the worst case clock cycle budget for a CTU (when it is in 16×16) is only 16. Only the IT2-Rec BIBO is enlarged, so that the transform path (i.e., CABAD-IQ-IT1-IT2-Rec) and the prediction path (i.e., CABAD-PDec-MC-Rec) of the system pipeline involve a comparative overall buffer size (in pixels) before merged in Rec. Compared with a baseline system pipeline that is built at the CTU level (such as in [36]) and adopts triple buffering to overcome the subpipeline delay, however, the proposed system pipeline based on BIBOs requires less pipeline buffers for pixels. Moreover, DBF and SAO are integrated into a single subpipeline [29]. This removes the necessity for dedicated pipeline buffers between them. To reduce the buffer size between Rec and DBF/SAO, the processing granularity of the latter is designed as half a CTU (i.e., 64×32) if the CTU size is 64×64. As shown in Fig. 11, the overall reduction of pixel buffer memory space is 61.9%. Also shown in Table III, in principle, each pixel buffer is organized as two banks of BIBOs to handle luma and chroma samples in parallel. The IT1–IT2 pixel buffer employs 32 banks for transposition, while the MC-Rec pixel buffer employs four banks for pixel format conversion.

E. Reduction of Parameter Buffer Space

In the proposed system pipeline, parameters are delivered using a network of FIFO queues. For a parameter FIFO at a certain level (CTU/CU/PU/TU/4×4 as the block size), one word of the FIFO is used to store the parameters of one block. Inside the FIFO, the blocks (words) are in the Z-scan order and no duplicated storage for the same CTU/CU/PU/TU/4×4 is necessary. Among the outputs of CABAD, TU-level parameters (TU coordinates, transform size and intra prediction modes, and so on) are required by many processing components. To avoid their duplicated storage in different FIFOs, the TU-level FIFOs are organized in a relay style. As shown in Fig. 12, the valid/ready signals of a processing component, an upstream FIFO, and a downstream FIFO are gated to each other, so that a parameter word is automatically moved from the upstream FIFO to the downstream FIFO at the same time it is consumed by the processing component. As shown in Fig. 2, the PU-level FIFOs for MV are organized in the same style for parameter delivery from PDec to FrmRd and MC. The sizing of FIFOs as shown in Fig. 13 involves negligible decoding speed degradation relative to the worst case sizing following the same baseline of CTU-level triple-buffering, as described in Section IV-D. In the meanwhile, overall parameter buffer size is reduced by 95.8%. The reason for the space reduction is twofold. On one hand, it is from the reduced granularities of the processing components, as described in Section IV-D, which reduces the worst case space requirements for parameters. On the other hand, with multiple buffering replaced by FIFOs, the buffer space no longer has to be sized only according to the worst cases (e.g., space for 256 TUs has to be preserved for a 64×64 CTU due to a minimum TU size of 4×4)—a congestion from insufficient FIFO space

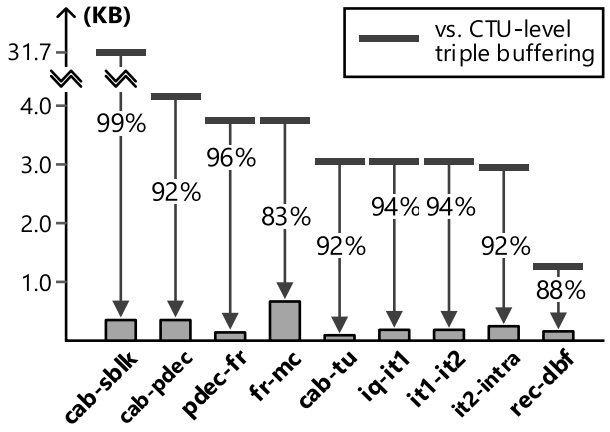


Fig. 13. Reduction of parameter buffer memory space.

TABLE IV
PIPELINE UTILIZATION OF PIXEL PROCESSING COMPONENTS

	Parallelism (pixel/cycle)	Utilization*
IQ	4×4	83.3%
IT1/IT2	4×4/8×2/16×1	83.3%
MC	8×4/4×4	41.7%
Rec	4×4	83.3%
DBF/SAO	4×4	83.3%

* At maximum throughput (4Gpixels/s LDP, decoder core running at 300MHz).

only results in a temporary pipeline stall that can be recovered until more stocks are consumed by the downstream processing components, rather than causing a decoding failure in the multiple buffering case.

F. Unified 4 × 4-Pixel/Cycle Parallelism

All processing components of the decoder are internally fully pipelined (called subpipeline), which does not necessarily need to be drained or refilled at the CTU/CU/PU/TU/4 × 4 boundaries. Only when a complete slice is finished, the decoder core is soft reset and the system pipeline emptied for the next slice. Most pixel processing components (IQ, IT1, IT2, Intra, Rec, DBF/SAO, and FrmWr) follow a unified parallelism of 4 × 4 pixels per clock cycle. MC is overdesigned to have a maximum output parallelism of 8 × 4 pixels per clock cycle to address the throughput variation from DRAM access and vertical MC interpolation. Theoretically, the 4 × 4-pixel/cycle parallelism can fulfill the throughput requirement of 8K 120 frames/s (4 Gpixels/s) decoding at 250 MHz. Considering various overheads including the fluctuation of DRAM traffic and CABAD bit rate, an excessive clock frequency of 300 MHz is used for the maximum specification, still resulting in a high pipeline utilization of 83.3%, as summarized in Table IV.

V. PROCESSING COMPONENTS

A. Inverse Transforms

The definition of HEVC ITs allows recursively decomposing large ITs into smaller ones, which enables a resource sharable hardware design. The straightforward implementation [18] of this decomposition led to a variable processing throughput depending on the TU size (e.g., N samples/cycle for N-point IT). To ensure a worst case performance, a line-based constant-throughput design was developed to process 4 × 1 samples per clock cycle [19]. The same design style, however, does not work for a higher parallelism, since the resulting line-based processing patterns (e.g., 16 × 1 samples/cycle) no longer apply to all TU sizes. In this paper, we address this issue with a multishape IT design. As shown in Fig. 14, the 1-D IT architecture is realized by extending a standard recursively decomposable 16-point IT core, which outputs 16×1, 8×1, and 4×1 samples for 16-, 8-, and 4-point ITs, respectively. By combining an eight-output engine for the odd portion of a 32-point IT, the architecture can also process one line of a 32-point transform (i.e., 32 × 1 samples) every other clock cycle. To align the throughput of all TU sizes, eight-point and two four-point cores are further added to generate additional 8 × 1 or 4 × 3 samples/cycle. As a result, the proposed 1-D IT architecture achieves a constant throughput of 16 samples/cycle, in 16 × 1, 16 × 1, 8 × 2, and 4 × 4, for 32-, 16-, 8-, and 4-point ITs, respectively. Through the reuse of 4-, 8-, and 16-point IT cores as the even portions of 8-, 16-, and 32-point cores, 24.5% area savings are also achieved.

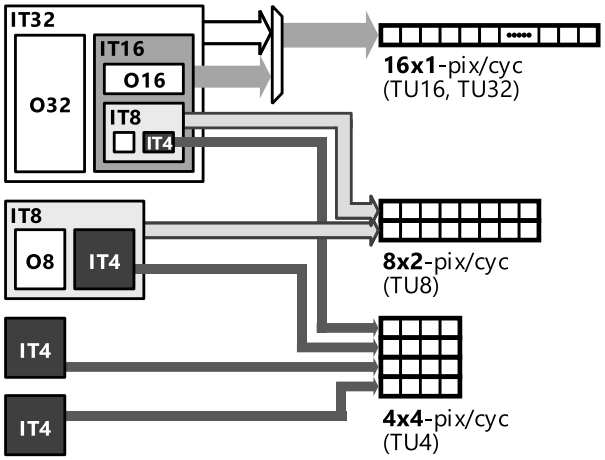


Fig. 14. 16-pixel/cycle multiple-shape 1-D IT. ITn stands for an n-point IT, which can be reused as the even portion of a 2n-point IT. On stands for the odd portion of an n-point IT. The 4×4 inverse DST is implemented separately.

The complete 2-D IT architecture is realized in a luma-chroma-parallel and row-column-pipelined style. The chroma portion occupies a smaller area by not supporting 32-point IT. Between the row (IT1) and column (IT2) ITs, the BIBO queue is divided into 16 (32 for both luma and chroma) banks. As shown in Fig. 15, owing to BIBO's random addressability, a cyclic memory mapping can be used inside each block of the BIBO to perform the required transposition between IT1 and IT2. While the BIBO queues are primarily used as pipeline buffers, the necessity for dedicated transposition memory is also removed.

B. Integration of Processing Components

Table II shows the pipeline granularity of primary processing components. CABAD is only interfaced with FIFO

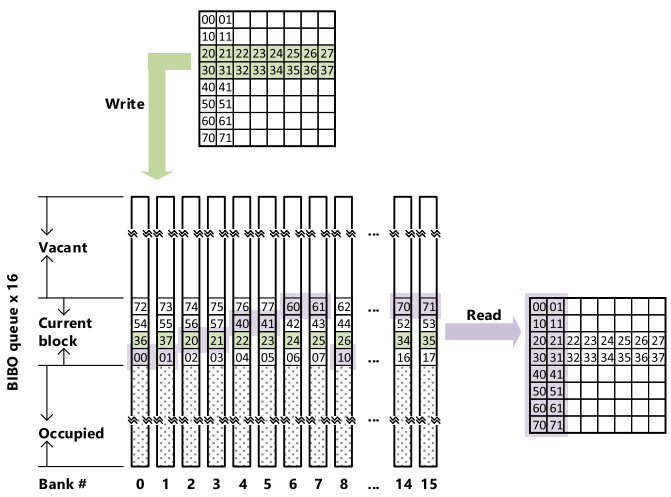


Fig. 15. BIBOs organized as a transposition buffer. This example shows how an 8 × 8 block is mapped. 16 × 16 and 32 × 32 blocks can be mapped in a similar style.

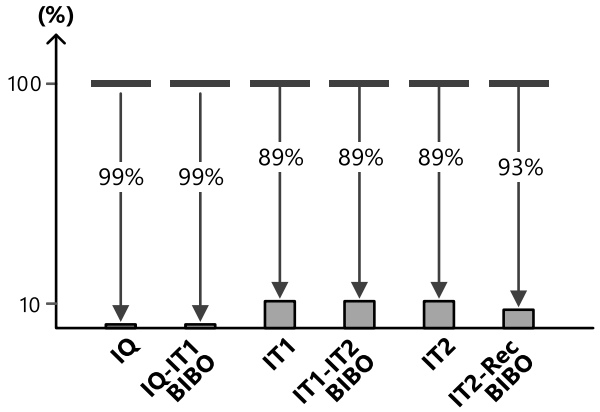


Fig. 16. Activity reduction for IQ, IT, and the related BIBOs. The experiments were performed over Class A sequences under LDP and RA configurations.

queues with valid/ready handshake. The transform path (CABAD-IQ-IT1-IT2-Rec) of the decoder involves many all-zeros, which provides chances for dynamic power reduction. With CABAD delivering coefficients of only nonzero 4 × 4 subblocks, unnecessary switching activities are skipped in IQ. Based on the TU-level parameters, operations for all-zero TUs are bypassed in IT1 and IT2. With the zero map function, memory write and read in BIBOs for all zeros are further skipped at the word level. As a result, an 88.7%–98.5% activity reduction in IQ, IT1/IT2, and the related BIBOs is achieved relative to performing full operations for all the input data including zeros, as summarized in Fig. 16. The related portion composes 33% of the logic area of the chip. Intra works at a TU rather than PU-based granularity to address the data dependencies at TU boundaries. MC is overdesigned to have an output throughput of 8 × 4 pixels/cycle for PUs wider than eight pixels. For 4×8 and 4×16 PUs, the output throughput is 4×4 pixels/cycle. Different from the six-tap interpolation filters in H.264, new 7/8-tap filters are applied in HEVC, which influences the

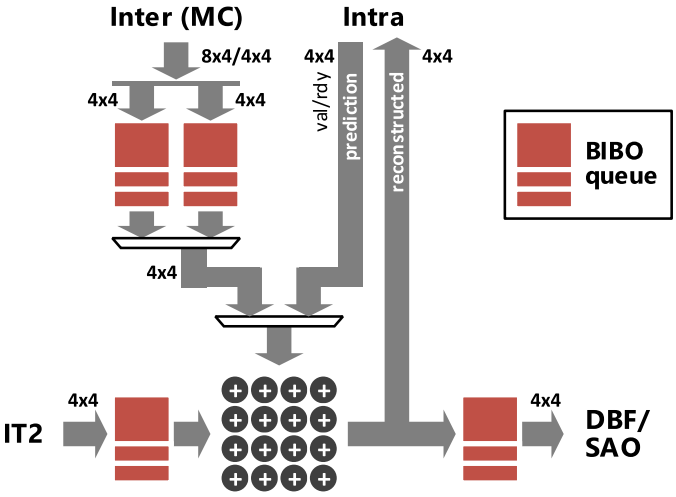


Fig. 17. Data flow of the luma portion of Rec. The chroma portion follows a similar structure.

MC memory design. In this chip, the data memory of the MC cache is organized as eight cyclic banks of 1K × 120-b SRAM to output a maximum of 16 × 4 pixels/cycle, which supports the 8 × 4 pixel/cycle MC interpolation. Compared with a typical dual-bank design used in H.264 decoders, the proposed style saves cache area and read power by 10% and 32%, respectively, while achieving a hit rate of 68%. Note the design target of the chip was to support 8K decoding at 120 and 60 frames/s for unidirectional and bidirectional inter prediction, respectively. So the MC interpolation and cache were designed accordingly. In bidirectional prediction, the peak MC throughput is halved.

Fig. 17 shows the data flow of Rec, which interfaces IT2, Intra, MC, and DBF/SAO. Between MC and Rec, two BIBO queues for 4 × 4 subblocks with even and odd column indices are used to convert the data format from 8 × 4/4 × 4 to 4 × 4 samples. To address the immediate data dependencies of Intra, it is interfaced with Rec with a valid/ready handshake and no intermediate buffer. The reconstructed samples, which are the sum of inverse transformed residuals and either of the inter- and intrapredicted samples, are also immediately fed back to Intra for the prediction of the next block.

More details of low-level designs of the processing components in this chip can be found in [9] for CABAD, [12] for PDec, [27] for Intra, [7] for MC, and [29] for DBF/SAO.

VI. Implementation Results

A. Specification

The video decoder core was designed at the register transfer level in SystemVerilog and Verilog. A corresponding software model was built in C++ over HEVC Test Model 13.0 [43]. Simulation-based test was conducted for most components with a test set containing 25 8-b and 2 10-b video sequences over four configurations [Intra, low-delay-P (LDP), low-delay, and random access] under the HEVC common test conditions (CTCs) [44]. Eight non-CTC sequences at 4K and 8K resolutions were also tested.

TABLE V
CHIP SPECIFICATION

Technology	40nm LL CMOS
Supply voltage	1.0V core, 1.5V DDR3 I/O, 2.5V digital I/O
Die size	4.95×4.63 mm² (incl. DDR3 PHY, DLL and PLLs)
Package	288-pin BGA
Logic gates	2887K equivalent NAND2
On-chip memory	396KB SRAMs and register files
External memory	64-bit DDR3 SDRAM
Maximum pixel throughput	4Gpixels/s
Maximum resolution	7680×4320@120fps
Measured core power consumption (25°C) [3]	690mW@300MHz[1]/660MHz[2], 1.0V, 4320p@120fps LDP [4] 501mW@200MHz[1]/500MHz[2], 1.0V, 4320p@ 60fps RA [5] 305mW@150MHz[1]/400MHz[2], 0.9V, 4320p@ 60fps LDP [4] 246mW@100MHz[1]/400MHz[2], 0.9V, 2160p@120fps RA [6]

[1] Clock frequency of decoder core.
[2] Clock frequency of DRAM.
[3] Power of decoder core, DDR3 controller and the digital portion of DDR3 PHY.
[4] 7680×4320 "NebutaFestival1" low-delay-P 116.4Mbps.
[5] 7680×4320 "NebutaFestival1" random-access 71.1Mbps.
[6] 3840x2160 "CrowdRun" random-access 35.8Mbps.

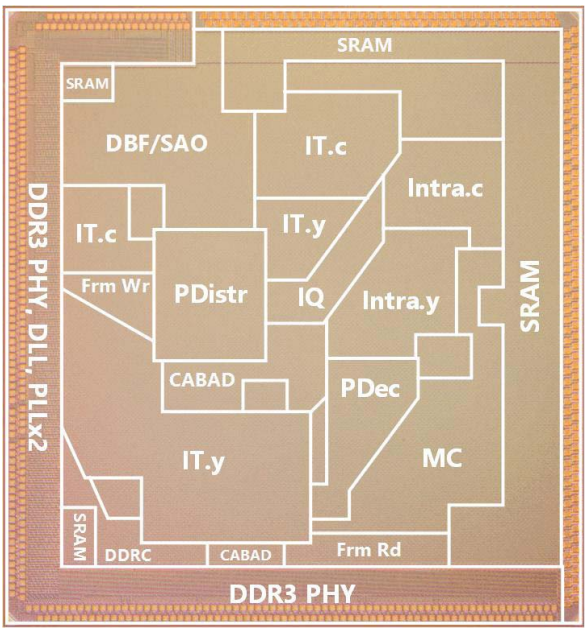


Fig. 18. Chip photo. Main parts of the chip components are as labeled. The unlabeled area consists of small fractions of these components generated in automatic placement and routing.

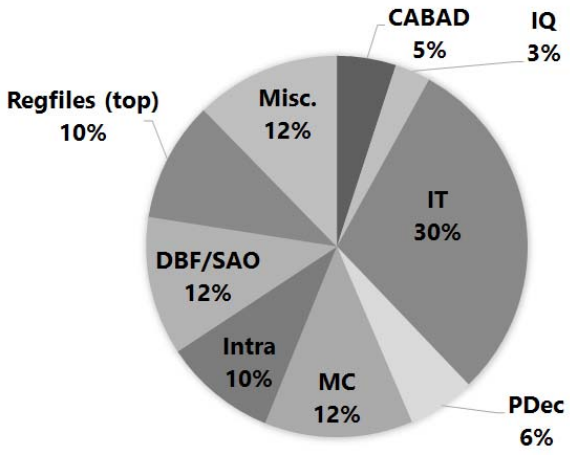


Fig. 19. Breakdown of utilization of logic gates (equivalent NAND2).

A test chip was fabricated in 40-nm CMOS process. Table V and Fig. 18 show the chip specification and micrograph, respectively. Along with the decoder core, a 64-b DDR3 PHY, a DLL, two PLLs for the decoder and DRAM clock domains, and I/O's were integrated into the 4.95×4.63 mm^2 die. The decoder core includes 2887K automatically placed and routed logic gates (equivalent NAND2). Fig. 19 shows the architecture and logic breakdown of key processing components.

The chip also includes 93 SRAM and register file instances generated by memory compilers, with an overall size of 396 kB. In addition to the BIBO queues (63 kB), the majority of on-chip memory usage is by the MC cache (131 kB), DBF/SAO (95 kB), PDec (29 kB), and Intra (25 kB). Especially, DBF/SAO consumes 80-kB memory for buffering five lines of pixel data, including five lines of luma and three lines of Cb/Cr samples. To support 10-b coding, all memory instances for pixel storage are 25% larger than their 8-b only version. To meet timing and implementation constraints, large logical buffers are decomposed into smaller memory cells, with bit size ranging from 8 to 128 and word size ranging from 128 to 2048. The slowest among these cells is a 256×128-b two-port (1R1W) register file with a minimum clock period of 3.14 ns (125° C and 0.99V), which is still fast enough for the required clock rate of 300 MHz.

The chip size was determined by the large pin count rather than the transistor count, which explains the relatively low

TABLE VI
COMPARISON WITH RECENT HEVC AND H.264 VIDEO DECODER CHIPS

	This work [51]	**ESSCIRC'14** [36]	**ISSCC'13** [45]	**A-SSCC'13** [34]	**ISSCC'12** [46]	**VLSI'10** [47]	**ISSCC'10** [42]
Video format(s)	HEVC	HEVC + multi-format	HEVC WD4	HEVC	H.264	H.264	H.264
Max. throughput	4Gpixel/s	531Mpixel/s	249Mpixel/s	72Mpixel/s	2Gpixel/s	530Mpixel/s	212Mpixel/s
Max. resolution	4320p @120fps	2160p @60fps	2160p @30fps	1080p @35fps	4320p @60fps	2160p @60fps	2160p @24fps
Logic gates	2887K	3454K	715K	446K	1338K	662K	414K
On-chip SRAM	396KB	154KB	124KB	10.2KB	79.9KB	59.6KB	9.0KB
Technology	40nm/1.0V	28nm/0.9V	40nm/0.9V	90nm/1.0V	65nm/1.2V	90nm/1.0V	90nm/1.0V
Clock rate @ max. TP	300MHz	350MHz	200MHz	224MHz	340MHz	175MHz	210MHz
Core power @ max. TP	690mW	104mW	76mW	36.9mW	410mW	189mW	59.5mW
Core power per pixel	0.15nJ/pixel* 0.25nJ/pixel**	0.20nJ/pixel	0.31nJ/pixel	0.59nJ/pixel	0.21nJ/pixel	0.36nJ/pixel	0.28nJ/pixel
DRAM config.	64b DDR3	32b LPDDR3	32b DDR3	n/a	64b DDR2	64b DDR1	n/a

* LDP@0.9V. ** RA@1.0V.

area utilization. In addition to the pins for the 64-b DDR3 interface, also implemented were an 8-b bus for feeding test bit streams and configurations into the chip, as well as a 64-b digital test access port.

With the decoder core and DRAM interface working at 300 and 660 MHz, respectively, the chip delivers a maximum throughput of 4 Gpixels/s, supporting 8K (7680×4320p) 120 frames/s decoding for the HEVC LDP configuration. At this maximum throughput, the measured core power dissipation is 690 mW at a 1 V core supply. For the random-access (RA) configuration that involves bidirectional inter prediction and thus a much higher per-pixel DRAM bandwidth requirement, the chip is capable of 8K 60 frames/s decoding with 501 mW. For lower specifications, such as 8K 60 frames/s LDP or 4K 120 frames/s RA, the core supply can be downscaled to 0.9 V by taking advantage of a lower clock rate. But the clock rate of DDR3 has a nominal lower bound of 400 MHz, which cancelled the energy efficiency improvement from voltage scaling to a certain extent.

B. Comparison

As shown in Table VI, the proposed video decoder chip is the first integrated 8K and Gpixel/s-level decoder for HEVC, achieving 7.5–55 times higher throughput than prior works [34], [36], [45]. Even when compared with the lower-complexity H.264/AVC decoder chips [42], [46], [47], it is still at least two times faster under a unidirectional inter prediction (i.e., LDP for HEVC) and as fast under a bidirectional inter prediction (i.e., RA for HEVC). Fig. 20 shows a comparison with normalization and technology scaling. Owing to a high pipeline utilization and a high parallelism, this paper achieves 3.1–3.6 times better core area efficiency than previous HEVC decoders [34], [45], in spite of the larger absolute area. The proposed decoder involves a higher usage of on-chip memory, primarily due to the enlarged line buffers and MC cache

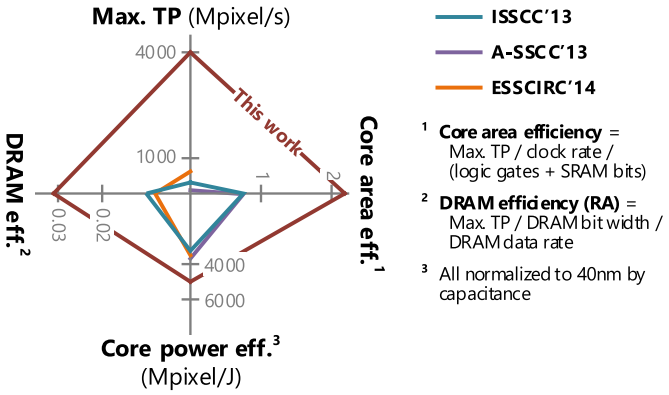


Fig. 20. Comparison on design efficiency. Two points not plotted due to lack of data or to avoid unfair comparison.

for addressing the higher resolution of 8K, and from the features not fully implemented in previous chips [34], [45], including the support for 10-b sampling, SAO, and the DDR3 DRAM interface. With a higher area efficiency and a comparative maximum clock rate, this paper naturally achieves 31%–55% superior energy efficiency than previous HEVC chips [36], [45].

C. DRAM Bandwidth Requirement

For this test chip, a relatively old-fashioned DDR3-1600 interface is applied. With this interface, the target performance of 8K 60/120 frames/s can be achieved in typical cases, primarily with the contributions from the MC cache [7] and a lossless reference frame recompression [48]. For product-level chips, state-of-the-art memory interfaces, such as DDR4-2400 (and its dual-channel version), wide-IO, and hybrid memory cube will be more suitable to secure the worst cases. However, even with advanced memory interfaces, the typical-case memory access optimizations will still be very useful in reducing the system power consumption.

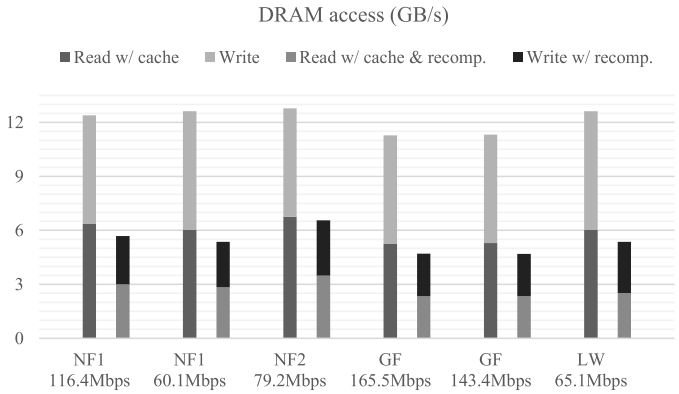


Fig. 21. Simulated DRAM traffic of the first ten frames of 8K LDP streams. NF1: NebutaFestival1. NF2: NebutaFesitval2. GF: Grass&Flowers. LW: LocomotiveWheel.

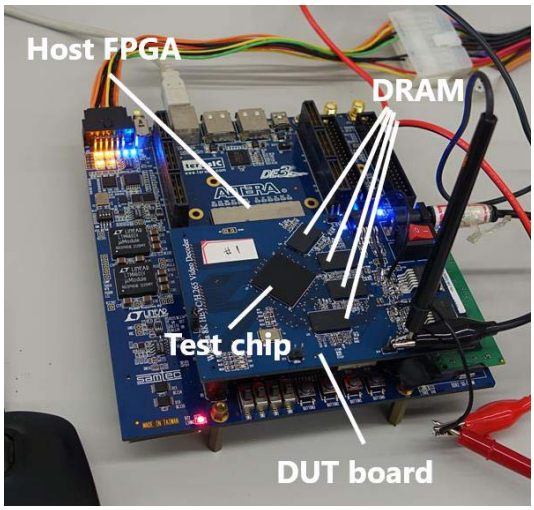


Fig. 22. Test boards.

In HEVC motion compensation, the reference block to be fetched for a current block is usually much larger than the latter, due to fractional MC interpolation and various alignment issues between the target pixels and their storage units. In the meanwhile, the reference blocks for adjacent current blocks significantly overlap with each other. The MC cache [7] is designed to reuse the overlapped data. As a result, the volume of the cache's reference frame read for decoding a typical LDP frame is only marginally larger than the frame size, as shown in Fig. 21.

The test chip also includes a lossless reference frame recompression [48], which enables compressing the frame data on-the-fly before they are written into the DRAM and reading the compressed data with random accessibility. With the recompression, overall DRAM access is reduced by 49%–59%, as shown in Fig. 21.

Though the reference frames are compressed, its target is only to reduce the volume of memory access instead of memory space. To ensure the worst case, regardless of the bit depth mode the decoder is working in, 256-b space of the DRAM is preserved for each 4×4 block with an original size of 192 and 240 b for 8- and 10-b modes, respectively. As a result, each 8K frame occupies 63.3 MB space of DRAM.

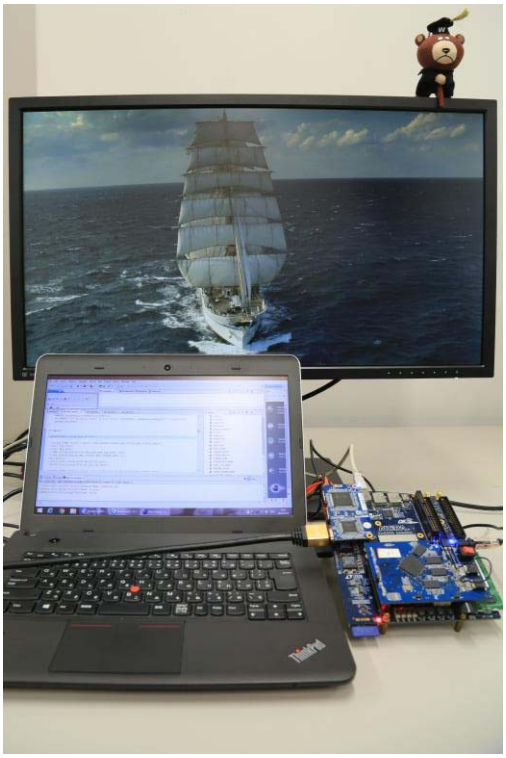


Fig. 23. Chip test and demo environment.

TABLE VII

DRAM SYSTEM POWER MODELING. ALL PARAMETERS NOT LISTED ARE SET AS DEFAULT. DRAM ACCESS STATISTICS ARE FROM SIMULATION OF THE FIRST TEN FRAMES OF NEBUTAFESTIVAL1 LDP 116.4 MB/s

DRAM Density	1Gb
Num. DQs per DRAM	×16
Speed grade	-125
System CK frequency	660MHz
Page hit rate	88.9%
% Read clock cycles	28.4%
% Write clock cycles	25.4%
Num. DRAMs	×4
Power per DRAM	299.7mW
Total DRAM power	**1198.8mW**

In practice, only part of the overall preserved space will be used due to the compact storage of the blocks after recompression. Since HEVC Level 6.2 defines a maximum decoded picture buffer size of six 8K frames, the overall DRAM space usage, including the current frame, reference frames, and colocated MVs, can be lower than 0.5 GB.

Power consumption of the DRAM is evaluated both through measurement and by modeling. The measured power is 969 mW from the 1.5 V power domain of the test board, which contains the analog part of the on-chip DDR3 PHY and the four external chips of 16-b 1-Gb SG-125 DDR3 from Micron. For modeling, a brief power calculator [49] is used. Table VII shows the modeling parameters generated from DRAM access statistics and the modeling results based on the specification of the same DDR3 chips used.

D. Test and Demo Environment

The chip was packaged in BGA and tested on a DUT board plugged into a Terasic DE3 motherboard with an Altera Stratix III host FPGA, as shown in Fig. 22. An NIOS II processor implemented in the FPGA was programmed to configure the test chip, supply bit streams, monitor the decoder status, and retrieve the results. A demo system, as shown in Fig. 23, was also established based on the test setup. The 64-b test access port of the chip was used to transfer the decoded pixels to the FPGA in real time. Due to its limited bandwidth, however, the decoded 8K video was first downsampled on the chip into a 1080p 4:2:2 format before retrieved by the FPGA and displayed.

VII. Conclusion

In this paper, a low memory-usage and high-utilization system pipeline design is presented for an 8K HEVC video decoder. In its realization, a novel BIBO queue building block plays an important role. A test decoder chip combining the proposed system pipeline with various component-level optimization achieves a significantly higher throughput than prior chips, while also demonstrating improved area and energy efficiency.

Though the proposed decoder design supports 8K 120 frames/s for unidirectional inter prediction, only 8K 60 frames/s is supported for bidirectional prediction, mainly due to the latter's huge DRAM bandwidth requirements. Moreover, CABAD is not yet pipelined with the pixel processing components at the slice or picture level (such as in [46] and [50]), which makes it highly sensible to the frame-level bit rate variation. Currently, the limited throughput of a single CABAD is also not enough to support 8K 120 frames/s decoding of all-intra HEVC streams, for which parallelism of multiple CABAD [46] at the frame or slice level is required. These remain problems to be addressed in the future.

References

[1] *High Efficiency Video Coding*, document ITU-T H.265/ISO/IEC 23008-2 HEVC, 2013.

[2] *Advanced Video Coding for Generic Audiovisual Services*, document ITU-T H.264/ISO/IEC 14496-10 AVC, 2005.

[3] J. Vanne, M. Viitanen, T. D. Hamalainen, and A. Hallapuro, "Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1885–1898, Dec. 2012.

[4] F. Sampaio, B. Zatt, M. Shafique, L. Agostini, J. Henkel, and S. Bampi, "Content-adaptive reference frame compression based on intra-frame prediction for multiview video coding," in *Proc. 20th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2013, pp. 1831–1835.

[5] L. Guo, D. Zhou, and S. Goto, "A new reference frame recompression algorithm and its VLSI architecture for UHDTV video codec," *IEEE Trans. Multimedia*, vol. 16, no. 8, pp. 2323–2332, Dec. 2014.

[6] X. Lian, Z. Liu, W. Zhou, and Z. Duan, "Lossless frame memory compression using pixel-grain prediction and dynamic order entropy coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 223–235, Jan. 2016.

[7] S. Wang, D. Zhou, J. Zhou, T. Yoshimura, and S. Goto, "VLSI implementation of HEVC motion compensation with distance biased direct cache mapping for 8K UHDTV applications," *IEEE Trans. Circuits Syst. Video Technol.*, doi: 10.1109/TCSVT.2015.2511858.

[8] J. Choi and Y. Choi, "High-throughput CABAC codec architecture for HEVC," *Electron. Lett.*, vol. 49, no. 18, pp. 1145–1147, Aug. 2013.

[9] Y. Zhao, J. Zhou, D. Zhou, and S. Goto, "A 610 Mbin/s CABAC decoder for H.265/HEVC level 6.1 applications," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Paris, France, Oct. 2014, pp. 1268–1272.

[10] J. Hahlbeck and B. Stabernack, "A 4k capable FPGA based high throughput binary arithmetic decoder for H.265/MPEG-HEVC," in *Proc. IEEE Int. Conf. Consum. Electron.-Berlin (IEEC-Berlin)*, Sep. 2014, pp. 388–390.

[11] Y. H. Chen and V. Sze, "A deeply pipelined CABAC decoder for HEVC supporting level 6.2 high-tier applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 5, pp. 856–868, May 2015.

[12] S. Wang, D. Zhou, J. Zhou, T. Yoshimura, and S. Goto, "Unified parameter decoder architecture for H.265/HEVC motion vector and boundary strength decoding," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E98-A, no. 7, pp. 1356–1365, Jul. 2015.

[13] E. Kalali and I. Hamzaoglu, "A low energy HEVC sub-pixel interpolation hardware," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 1218–1222.

[14] C. M. Diniz, M. Shafique, S. Bampi, and J. Henkel, "A reconfigurable hardware architecture for fractional pixel interpolation in high efficiency video coding," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 34, no. 2, pp. 238–251, Feb. 2015.

[15] W. Zhou, X. Zhou, and X. Lian, "An efficient interpolation filter VLSI architecture for HEVC," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 2015, pp. 1106–1110.

[16] H. Maich, G. Paim, V. Afonso, L. Agostini, B. Zatt, and M. Porto, "A multi-standard interpolation hardware solution for H.264 and HEVC," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 2910–2914.

[17] G. Pastuszak and M. Trochimiuk, "Architecture design of the high-throughput compensator and interpolator for the H.265/HEVC encoder," *J. Real-Time Image Process.*, vol. 11, no. 4, pp. 663–673, Apr. 2016.

[18] S. Shen, W. Shen, Y. Fan, and X. Zeng, "A unified forward/inverse transform architecture for multi-standard video codec design," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E96-A, no. 7, pp. 1534–1542, 2013.

[19] H. Sun, D. Zhou, P. Liu, and S. Goto, "A low-cost VLSI architecture of multiple-size IDCT for H.265/HEVC," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E97-A, no. 12, pp. 2467–2476, Dec. 2014.

[20] M. Tikekar, C.-T. Huang, V. Sze, and A. Chandrakasan, "Energy and area-efficient hardware implementation of HEVC inverse transform and dequantization," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 2100–2104.

[21] P. K. Meher, S. Y. Park, B. K. Mohanty, K. S. Lim, and C. Yeo, "Efficient integer DCT architectures for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 168–178, Jan. 2014.

[22] T. Dias, N. Roma, and L. Sousa, "Unified transform architecture for AVC, AVS, VC-1 and HEVC high-performance codecs," *EURASIP J. Adv. Signal Process.*, vol. 2014, no. 1, p. 108, 2014.

[23] Y. Fan, L. Huang, Y. Bai, and X. Zeng, "A parallel-access mapping method for the data exchange buffers around DCT/IDCT in HEVC encoders based on single-port SRAMs," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 62, no. 12, pp. 1139–1143, Dec. 2015.

[24] N. Zhou, D. Ding, and L. Yu, "On hardware architecture and processing order of HEVC intra prediction module," in *Proc. Picture Coding Symp. (PCS)*, Dec. 2013, pp. 101–104.

[25] Z. Liu, D. Wang, H. Zhu, and X. Huang, "41.7BN-pixels/s reconfigurable intra prediction architecture for HEVC 2560 × 1600 encoder," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2013, pp. 2634–2638.

[26] C.-T. Huang, M. Tikekar, and A. P. Chandrakasan, "Memory-hierarchical and mode-adaptive HEVC intra prediction architecture for quad full HD video decoding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 7, pp. 1515–1525, Jul. 2014.

[27] J. Zhou, D. Zhou, S. Wang, T. Yoshimura, and S. Goto, "High performance VLSI architecture of H.265/HEVC intra prediction for 8K UHDTV video decoder," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E98-A, no. 12, pp. 2519–2527, Dec. 2015.

[28] F. Amish and E.-B. Bourennane, "Fully pipelined real time hardware solution for high efficiency video coding (HEVC) intra prediction," *J. Syst. Archit.*, vol. 64, pp. 133–147, Nov. 2015.

[29] J. Zhu, D. Zhou, and S. Goto, "A high performance HEVC de-blocking filter and SAO architecture for UHDTV decoder," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E96-A, no. 12, pp. 2612–2622, Dec. 2013.

[30] M. Mody, N. Nandan, and T. Hideo, "High throughput VLSI architecture supporting HEVC loop filter for ultra HDTV," in *Proc. IEEE Int. Conf. Consum. Electron.-Berlin (IEEC-Berlin)*, Sep. 2013, pp. 54–57.

[31] M. Tomida, Y. Tanida, T. Song, and T. Shimamoto, "Small area VLSI architecture for deblocking filter of HEVC," in *Proc. IEEE Int. Conf. Consum. Electron.-Berlin (IEEC-Berlin)*, Sep. 2015, pp. 294–297.

[32] W. Cheng, Y. Fan, Y. Lu, Y. Jin, and X. Zeng, "A high-throughput HEVC deblocking filter VLSI architecture for 8k × 4k application," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 605–608.

[33] W. Zhou, J. Zhang, X. Zhou, Z. Liu, and X. Liu, "A high-throughput and multi-parallel VLSI architecture for HEVC deblocking filter," *IEEE Trans. Multimedia*, vol. 18, no. 6, pp. 1034–1047, Jun. 2016.

[34] C.-H. Tsai, H.-T. Wang, C.-L. Liu, Y. Li, and C.-Y. Lee, "A 446.6K-gates 0.55–1.2V H.265/HEVC decoder for next generation video applications," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2013, pp. 305–308.

[35] M. Tikekar, C.-T. Huang, C. Juvekar, V. Sze, and A. P. Chandrakasan, "A 249-Mpixel/s HEVC video-decoder chip for 4K ultra-HD applications," *IEEE J. Solid-State Circuits*, vol. 49, no. 1, pp. 61–72, Jan. 2014.

[36] C.-C. Ju *et al.*, "A 0.2 nJ/pixel 4K 60 fps Main-10 HEVC decoder with multi-format capabilities for UHD-TV applications," in *Proc. Eur. Solid-State Circuits Conf. (ESSCIRC)*, Sep. 2014, pp. 195–198.

[37] P.-T. Chiang *et al.*, "A QFHD 30-frames/s HEVC decoder design," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 4, pp. 724–735, Apr. 2016.

[38] D. Engelhardt, J. Moller, J. Hahlbeck, and B. Stabernack, "FPGA implementation of a full HD real-time HEVC main profile decoder," *IEEE Trans. Consum. Electron.*, vol. 60, no. 3, pp. 476–484, Aug. 2014.

[39] M. Abeydeera, M. Karunaratne, G. Karunaratne, K. De Silva, and A. Pasqual, "4K real-time HEVC decoder on an FPGA," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 236–249, Jan. 2016.

[40] C.-C. Ju *et al.*, "A 125 Mpixels/sec full-HD MPEG-2/H.264/VC-1 video decoder for Blu-ray applications," in *Proc. IEEE Asian Solid-State Circuits Conf.*, Nov. 2008, pp. 9–12.

[41] K. Iwata *et al.*, "A 256 mW 40 Mbps full-HD H.264 high-profile codec featuring a dual-macroblock pipeline architecture in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1184–1191, Apr. 2009.

[42] T.-D. Chuang *et al.*, "A 59.5 mW scalable/multi-view video decoder chip for quad/3D full HDTV and video streaming applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2010, pp. 330–331.

[43] *HEVC Test Model (HM) 13.0*. [Online]. Available: https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags/HM-13.0

[44] F. Bossen, *Common Test Conditions and Software Reference Configurations*, document JCTVC-H1100, 2012.

[45] C.-T. Huang, M. Tikekar, C. Juvekar, V. Sze, and A. Chandrakasan, "A 249 Mpixel/s HEVC video-decoder chip for quad full HD applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2013, pp. 162–164.

[46] D. Zhou, J. Zhou, J. Zhu, P. Liu, and S. Goto, "A 2 Gpixel/s H.264/AVC HP/MVC video decoder chip for Super Hi-Vision and 3DTV/FTV applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2012, pp. 224–225.

[47] D. Zhou *et al.*, "A 530 Mpixels/s 4096 × 2160@60 fps H.264/AVC high profile video decoder chip," in *Proc. Symp. VLSI Circuits (VLSI)*, Honolulu, HI, USA, 2010, pp. 171–172.

[48] D. Zhou, L. Guo, J. Zhou, and S. Goto, "Reducing power consumption of HEVC codec with lossless reference frame recompression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Paris, France, Oct. 2014, pp. 2120–2124.

[49] *DDR3 SDRAM System-Power Calculator*, Micron, Boise, ID, USA, 2011.

[50] D. Zhou *et al.*, "A 530 Mpixels/s 4096 × 2160@60 fps H.264/AVC high profile video decoder chip," *IEEE J. Solid-State Circuits*, vol. 46, no. 4, pp. 777–788, Apr. 2011.

[51] D. Zhou *et al.*, "A 4 Gpixel/s 8/10 b H.265/HEVC video decoder chip for 8K ultra HD applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Jan./Feb. 2016, pp. 266–267.

Dajiang Zhou (S'08–M'10) received the B.E. and M.E. degrees from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degree in engineering from Waseda University, Kitakyushu, Japan, in 2010.

He was a Researcher and an Assistant Professor with Waseda University. He is currently with the Ant Financial Services Group, Hangzhou, China, as a Senior Technical Expert in multimedia algorithms. His current research interests include algorithms and implementations for signal processing and artificial intelligence, especially in low-power high-performance architectures for video codecs and neural networks.

Dr. Zhou was a recipient of the research fellowship of the Japan Society for the Promotion of Science from 2009 to 2011. He received a number of awards, including the Best Student Paper Award of VLSI Circuits Symposium 2010, the International Low Power Design Contest Award of ACM ISLPED 2010, the 2013 Kenjiro Takayanagi Young Researcher Award, the 2014 Waseda Research Award (International Research Impact Sector), the 2014 Best Journal Paper Award of IEICE, and the Chinese Government Award for Excellent Students Abroad of 2010. He also received the 2012 Semiconductor of the Year Award of Japan.

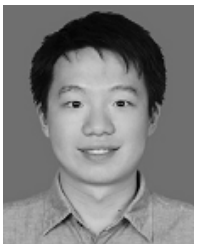
Shihao Wang received the B.E. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2012, and the M.E. degree in engineering from Waseda University, Kitakyushu, Japan, in 2014, where he is currently pursuing the Ph.D. degree in engineering.

His current research interests include VLSI architectures and algorithms for multimedia signal processing and neural networks.

Dr. Wang was a recipient of the Ting Hsin Scholarship in 2013 and 2014. He is being supported by the Graduate Program of Embodiment Informatics held by Waseda University, Kitakyushu, Japan, and Ministry of Education, Culture, Sports, Science and Technology, Tokyo, Japan.

Heming Sun received the B.E. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2011, the M.E. degree from Waseda University, Kitakyushu, Japan, in 2012, and the M.E. degree from Shanghai Jiao Tong University in 2014. He is currently pursuing the Ph.D. degree at Waseda University.

His current research interests include algorithms and VLSI architectures for multimedia signal processing.



Jianbin Zhou received the B.E. degree in software engineering from the South China University of Technology, Guangzhou, China, in 2012, and the M.E. degree in engineering from Waseda University, Kitakyushu, Japan, in 2014, where he is currently pursuing the Ph.D. degree.

His current research interests include VLSI architectures and algorithms for multimedia signal processing.

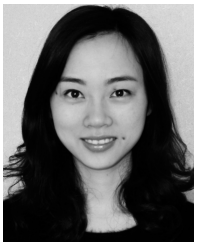
Mr. Zhou was a recipient of the Ting Hsin Scholarship in 2013 and 2014.

Jiayi Zhu received the B.E. and M.E. degrees from Shanghai Jiao Tong University, Shanghai, China. He is currently pursuing the Ph.D. degree at Waseda University, Kitakyushu, Japan.

His current research interests include algorithms and VLSI architectures for multimedia and communication signal processing.

Yijin Zhao received the B.E. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2013, the M.E. degree from Waseda University, Kitakyushu, Japan, in 2014, and the M.E. degree from Shanghai Jiao Tong University in 2016.

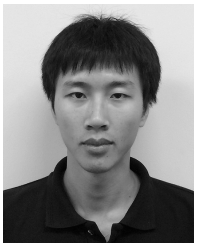
She is currently with Baidu, Inc., Beijing, China. Her current research interests include VLSI architectures and algorithms for multimedia signal processing.



Jinjia Zhou (S'12–M'13) received the B.E. degree from Shanghai Jiao Tong University, Shanghai, China, in 2007, and the M.E. and Ph.D. degrees from Waseda University, Kitakyushu, Japan, in 2010 and 2013, respectively.

She was a Researcher with Waseda University from 2013 to 2016. She is currently an Associate Professor with Hosei University, Tokyo, Japan. Her current research interests include algorithms and VLSI architectures for video coding.

Dr. Zhou received the research fellowship of the Japan Society for the Promotion of Science from 2010 to 2013. She was a recipient of the Chinese Government Award for Excellent Students Abroad of 2012.



Shuping Zhang received the B.E. degree from the Beijing Institute of Technology, Beijing, China, in 2011, and the M.E. degree from Waseda University, Kitakyushu, Japan, in 2014, where he is currently pursuing the Ph.D. degree.

His current research interests include 3-D integration implementations for video coding applications.

Shinji Kimura (M'85) received the B.E., M.E., and D.Eng. degrees in information science from Kyoto University, Kyoto, Japan, in 1982, 1984, and 1989, respectively.

He was a Visiting Scientist with Carnegie Mellon University, Pittsburgh, PA, USA, from 1989 to 1990, and a Visiting Scholar with Stanford University, Stanford, CA, USA, from 2000 to 2001. He has been an Assistant Professor with Kobe University, Kobe, Japan, since 1985, and an Associate Professor with the Nara Institute of Science and Technology, Ikoma, Japan, since 1993. He has been a Professor with Waseda University, Kitakyushu, Japan, since 2002. His current research interests include the formal and timing verification of logic circuits, the hardware/software co-design methodologies, reconfigurable hardware, and the low-power design.

Dr. Kimura is a Fellow of IEICE and a member of the Information Processing Society of Japan and the IEEE Computer Society. He has served as an Executive Committee Member of ICCAD 2011 and 2012, and the General Chair of ASP-DAC 2013.

Takeshi Yoshimura (M'86) received the B.E., M.E., and D.Eng. degrees from Osaka University, Osaka, Japan, in 1972, 1974, and 1997, respectively.

He joined NEC Corporation, Kawasaki, Tokyo, Japan, in 1974, where he was involved with research and development efforts devoted to computer application systems for communication network design, hydraulic network design, and VLSI computer-aided design (CAD). From 1979 to 1980, he was with the Electronics Research Laboratory, University of California Berkeley, CA, USA, where he was involved with VLSI CAD layout. He is currently a Professor with Waseda University, Kitakyushu, Japan.

Dr. Yoshimura is a member of the Information Processing Society of Japan. He received the Best Paper Awards from the Institute of Electronics, Information and Communication Engineers of Japan, and the IEEE CAS Society.

Satoshi Goto (S'69–M'77–SM'84–F'86–LF'11) received the B.E. and M.E. degrees in electronics and communication engineering and D.Eng. degree from Waseda University, Kitakyushu, Japan, in 1968, 1970, and 1981, respectively.

He joined NEC Laboratories in 1970, where he was involved with LSI design, multimedia systems, and software as general manager and vice president. From 2003, he was a Professor with the Graduate School of Information, Production and Systems, Waseda University, where he is currently a Professor Emeritus. He has authored seven books and over 300 technical papers in international journals and conferences. His current research interests include VLSI design methodologies for multimedia and mobile applications.

Dr. Goto is a Fellow of IEICE and a member of the Science Council of Japan. He was a Board Member of the IEEE Circuits and Systems (CAS) Society. He served as the General Chair of ICCAD and ASPDAC. He received a number of awards and honors, including the Distinguished Achievement Awards from IEICE and the Jubilee Medal from the IEEE.