8K超高精細動画像用H.265/HEVCデコーダLSI

周　大江
早稲田大学理工学術院情報生産システム研究科
〒８０８－０１３５　福岡県北九州市若松区ひびきの２－７

An 8K Ultra HD Video Decoder Chip for H.265/HEVC

Dajiang Zhou
Graduate School of Information, Production and Systems, Waseda University
2-7 Hibikino, Kitakyushu, Fukuoka 808-0135, Japan

1. Introduction

8K Ultra HD is being promoted as the next-generation digital video format. From a communication channel perspective, the latest High Efficiency Video Coding standard (H.265/HEVC) greatly enhances the feasibility of 8K by a doubled compression ratio. Implementation of source codecs, however, is challenged by the multiplication of an ultra-high throughput requirement and an increased complexity per pixel. The former factor corresponds to up to 10 bit/pixel, 7680×4320 pixel/frame and 120 frame/second, overall 80× of 1080p HD. The latter comes from the new features of HEVC relative to its predecessor H.264/AVC. The most challenging of them is the enlarged and highly variable-size coding/prediction/transform units (CU/PU/TU), which significantly increase 1) the requirement for on-chip memory as pipeline buffers, 2) the difficulty in ensuring pipeline utilization, and 3) the complexity of inverse transforms (IT).

This paper presents the first HEVC decoder chip supporting 8K Ultra HD [6], featuring a 16-pixel/cycle true-variable-block-size system pipeline. The pipeline 1) saves on-chip memory with a novel block-in-block-out (BIBO) queue system and a parameter delivery network, and 2) allows high design efficiency and utilization of processing components through local synchronization. Key optimizations on the component level are also presented.

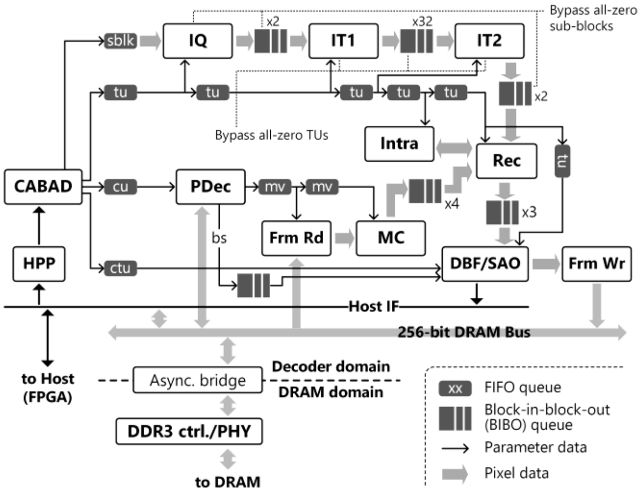


Figure 1: Chip block diagram.

2. Proposed Architecture

Fig. 1 shows the system architecture. The chip comprises two primary clock domains: the decoder domain consisting of processing components from High-level Parameter Parser (HPP) to Frame Writer, and the DRAM domain consisting of the DRAM controller and PHY. Parameter Decoder (PDec), Frame Reader and Frame Writer share DRAM bandwidth through a 256-bit bus, which is connected to the DRAM controller through an asynchronous bridge. In the DRAM domain, the controller and PHY follows a 1:2 clock ratio, so a 400MHz controller can support a maximum data rate of DDR3-1600.

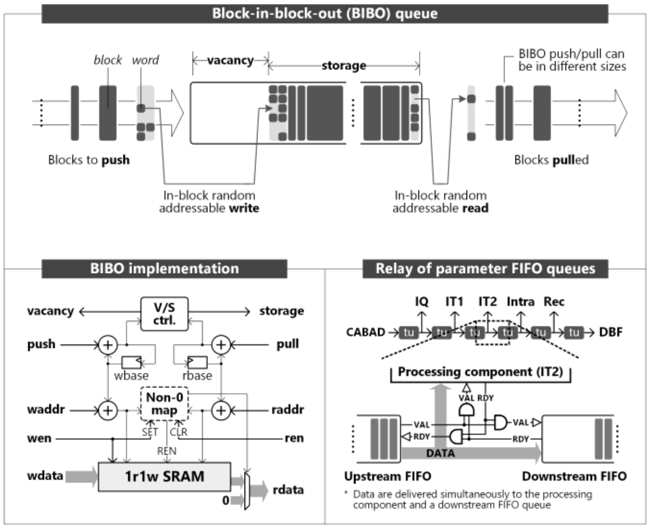

Block-in-block-out (BIBO) queue
block
word
vacancy
storage
BIBO push/pull can be in different sizes
Blocks to push
Blocks pulled
In-block random addressable write
In-block random addressable read
BIBO implementation
vacancy
V/S ctrl.
storage
push
pull
wbase
rbase
waddr
Non-0 map
raddr
wen
SET
CLR
ren
REN
wdata
1r1w SRAM
0
rdata
Relay of parameter FIFO queues
IQ
IT1
IT2
Intra
Rec
CABAD
tu
DBF
Processing component (IT2)
VAL RDY
VAL
RDY
DATA
Upstream FIFO
Downstream FIFO
* Data are delivered simultaneously to the processing component and a downstream FIFO queue

Figure 2: Block-in-block-out (BIBO) queue and relay of parameter FIFO queues.

Fig. 2 describes the BIBO queue for buffering pixel data. A BIBO queue combines the features of a queue and an array by allowing random addressing of *words* inside a *block* while storing the *blocks* in a first-in-first-out manner. *Blocks* can be in a variable size specified by the BIBO's writer/reader client by giving a push/pull size together with the write/read of the last *word* of the *block*. In implementation, BIBO translates the in-block address given by a client to the real address of a wrapped SRAM, by maintaining a base address for each of the writer and reader sides. Vacancy and storage levels are also maintained so that a client can judge whether the required space or data are available by comparing the level with current block size. Push and pull can be in different sizes even for the same *words*. *Blocks* can be combined and split seamlessly between writer and reader by addressing the pixel data in a Z-scan order. For power saving, the BIBO can also maintain a non-0 map of register bits set/cleared at each write/read cycle. With this feature writing of zeros can be skipped while the corresponding reading is bypassed by the BIBO.

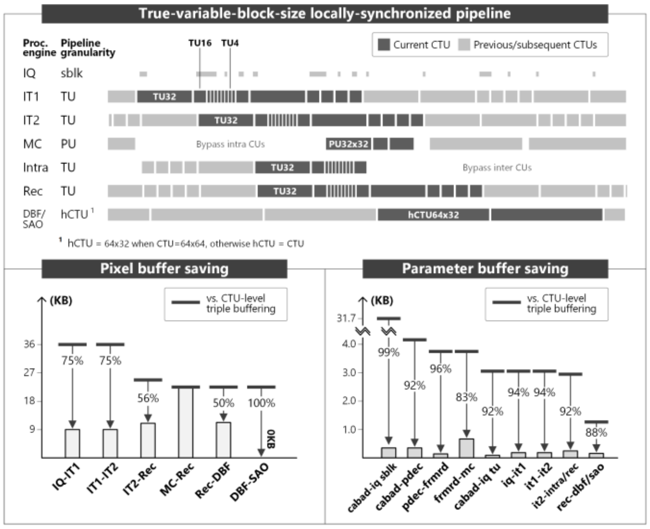


Figure 3: Proposed system pipeline.

Fig. 3 shows the system pipeline. H.264 decoders [1, 2] usually pipeline processing components on a unified Macroblock granularity. In HEVC, with Macroblocks replaced by enlarged and more hierarchical Coding Tree Units (CTU), such a design style suffers from huge requirement for pipeline buffers. The increased number of CU/PU/TU sizes also significantly increases the cases that have to be considered, leading to difficulty in efficiently controlling and interfacing the processing components. We propose a new pipeline to address these problems. In the pipeline, parameters from CABAD are classified into four levels of CTU, CU, TU and sub-block (4×4), and distributed through a network of FIFO queues. Motion vectors

(MV) from PDec are delivered in the same way on the PU level. Pixel data are buffered using BIBOs. The pipeline has the following features. 1) It is synchronized locally using the vacancy/storage status of FIFO and BIBO queues, rather than by a global scheduler. This allows IQ, IT/Intra, PDec and MC to be pipelined in different granularity: by sub-block, TU, CU and PU, respectively. This processing mechanism is the most natural for the components' algorithms, which enables efficient implementation. 2) For each component, processing granularity is also variable following the current CU/PU/TU size, while combination and splitting of blocks in different sizes are automated by BIBOs. 3) Most components' interfaces are simplified by handshaking with only FIFO and BIBO queues. 4) Pixel buffers are reduced by sizing BIBO queues according to the least common multiple of writer and reader granularity, rather than CTUs. An overall 61.9% saving of pixel buffers is achieved by further removing the buffer between Deblocking Filter (DBF) and SAO with an integrated sub-pipeline. 5) Parameter buffers are saved by sizing the FIFO queues according to relatively pessimistic (e.g. assuming average TU size as 8×8) rather than the worst (e.g. assuming all 4×4 TUs) cases. By further optimizing the TU and MV queues in a relay structure (Fig. 2), overall parameter buffers are saved by 95.8%. 6) Primary pixel processing components including IQ, IT, Intra, Rec and DBF/SAO are designed in a common parallelism of 4×4-pixel/cycle. The rest components are designed to match the same target throughput. Overall pipeline utilization achieves 83.3% despite overheads mainly from DRAM and CABAD.

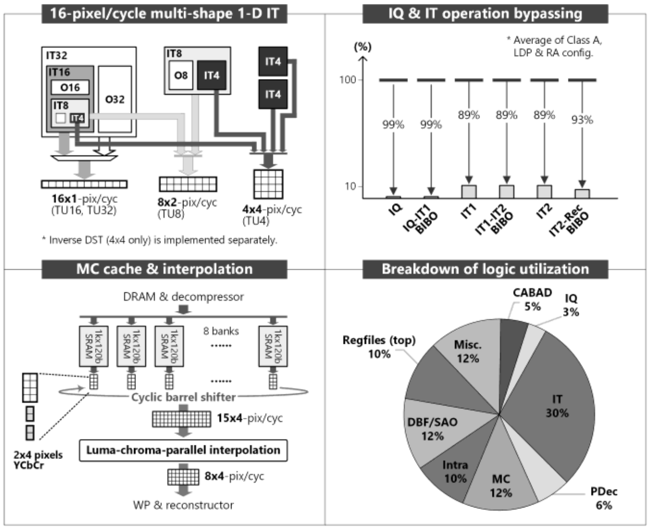
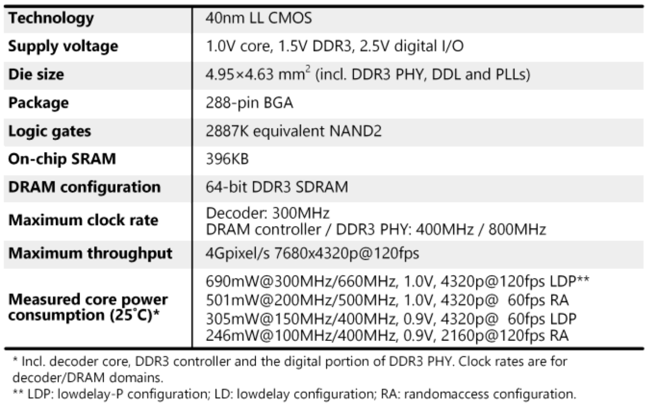


Figure 4: Component-level optimizations and breakdown of logic utilization.

Fig. 4 shows the architecture and logic breakdown of key processing components. For a highly parallel IT, conventional line-based processing patterns (e.g. 16×1-pixel/cycle) of recursive decomposition architectures no longer apply to all TU sizes. We propose a multi-shape 1-D IT to address this issue. Processing patterns of 4×4-, 8×2- and 16×1-pixel/cycle are applied for TU4, TU8 and above, respectively, to ensure a constant 16-pixel/cycle throughput. Meanwhile 24.5% logic area is saved by reusing the even portions of IT32, IT16 and IT8. Complete 2-D IT is realized in a luma-chroma-parallel and row-column-pipelined style. Pipeline buffers between IT1 and IT2 are organized as 32 banks of BIBO queues for transposition. To save power consumption, all-zero blocks are bypassed on TU and sub-block levels, resulting in 88.7% to 98.5% activity reduction in IQ, IT and the related BIBOs. The data memory of MC cache is organized as 8 banks of 1K×120-bit SRAMs to ensure the data preparation of an 8×4-pixel/cycle interpolation, overdesigned due to MC's variable throughput nature.

Compared to a dual-bank design, the proposed one saves cache memory's physical area and read power by 10% and 32%, respectively, while achieving a hit rate of 68%. Together with other DRAM access optimizations including lossless frame recompression and DRAM mapping [2], overall DRAM access cycles are saved by over 80%.



Technology	40nm LL CMOS
Supply voltage	1.0V core, 1.5V DDR3, 2.5V digital I/O
Die size	4.95×4.63 mm^2 (incl. DDR3 PHY, DDL and PLLs)
Package	288-pin BGA
Logic gates	2887K equivalent NAND2
On-chip SRAM	396KB
DRAM configuration	64-bit DDR3 SDRAM
Maximum clock rate	Decoder: 300MHz DRAM controller / DDR3 PHY: 400MHz / 800MHz
Maximum throughput	4Gpixel/s 7680x4320p@120fps
Measured core power consumption (25˚C)*	690mW@300MHz/660MHz, 1.0V, 4320p@120fps LDP** 501mW@200MHz/500MHz, 1.0V, 4320p@ 60fps RA 305mW@150MHz/400MHz, 0.9V, 4320p@ 60fps LDP 246mW@100MHz/400MHz, 0.9V, 2160p@120fps RA

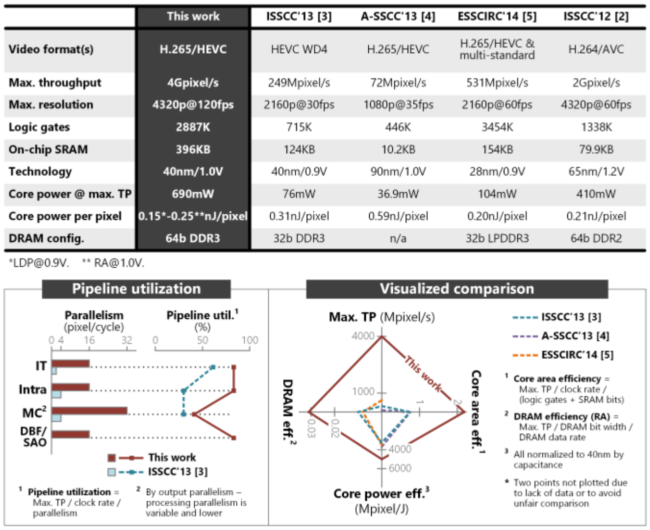
* Incl. decoder core, DDR3 controller and the digital portion of DDR3 PHY. Clock rates are for decoder/DRAM domains.
** LDP: lowdelay-P configuration; LD: lowdelay configuration; RA: randomaccess configuration.

Figure 5: Chip specification and measurement results.

3. Implementation Results

Fig. 5 gives the specification of the chip [6] in 40nm. Its die size is 22.92mm^2 including a 64-bit DDR3 PHY, DLL, PLLs, and the digital core comprising 2887K logic gates and 396KB on-chip SRAM. With a core supply of 1.0V and the decoder/DRAM domains working at 300MHz/660MHz, 4Gpixel/s decoding is achieved for HEVC lowdelay-P configuration, consuming 690mW core power. At 200MHz/500MHz, 2Gpixel/s is achieved for the randomaccess configuration.

Fig. 6 gives the comparison to prior arts [2-5]. The proposed design supports 7680×4320@120fps decoding, 7.5× to 55× faster than previous works [3-5]. Though logic gate count and SRAM are also larger, normalization and technology scaling shows 3.1× to 3.6× better area efficiency and 31% to 55% better energy efficiency, with a significant portion of the improvement contributed by the 10.6% to 52.2% higher pipeline utilization relative to [3]. The larger SRAM usage is primarily from enlarged line buffers and MC cache for addressing the features of 8K, and from features not implemented in previous chips [3-4] including 10-bit sampling, SAO and the DRAM interface. Our chip also achieves 3.2× to 3.6× better efficiency than [3, 5] in utilizing DRAM.



	This work	ISSCC'13 [3]	A-SSCC'13 [4]	ESSCIRC'14 [5]	ISSCC'12 [2]
Video format(s)	**H.265/HEVC**	HEVC WD4	H.265/HEVC	H.265/HEVC & multi-standard	H.264/AVC
Max. throughput	**4Gpixel/s**	249Mpixel/s	72Mpixel/s	531Mpixel/s	2Gpixel/s
Max. resolution	**4320p@120fps**	2160p@30fps	1080p@35fps	2160p@60fps	4320p@60fps
Logic gates	**2887K**	715K	446K	3454K	1338K
On-chip SRAM	**396KB**	124KB	10.2KB	154KB	79.9KB
Technology	**40nm/1.0V**	40nm/0.9V	90nm/1.0V	28nm/0.9V	65nm/1.2V
Core power @ max. TP	**690mW**	76mW	36.9mW	104mW	410mW
Core power per pixel	**0.15*-0.25**nJ/pixel**	0.31nJ/pixel	0.59nJ/pixel	0.20nJ/pixel	0.21nJ/pixel
DRAM config.	**64b DDR3**	32b DDR3	n/a	32b LPDDR3	64b DDR2

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

Pipeline utilization
Parallelism (pixel/cycle)
0
4
16
32
Pipeline util.[1] (%)
0
50
100
IT
Intra
MC[2]
DBF/SAO
This work
ISSCC'13 [3]
[1] Pipeline utilization = Max. TP / clock rate / parallelism
[2] By output parallelism – processing parallelism is variable and lower
Visualized comparison
Max. TP (Mpixel/s)
4000
1000
This work
Core area eff.[1]
1
2
DRAM eff.[2]
0.03
0.02
Core power eff.[3] (Mpixel/J)
4000
6000
ISSCC'13 [3]
A-SSCC'13 [4]
ESSCIRC'14 [5]
[1] Core area efficiency = Max. TP / clock rate / (logic gates + SRAM bits)
[2] DRAM efficiency (RA) = Max. TP / DRAM bit width / DRAM data rate
[3] All normalized to 40nm by capacitance
* Two points not plotted due to lack of data or to avoid unfair comparison

Figure 6: Chip comparison.

4. Acknowledgements

The author would like to express his sincere gratitude to Kenjiro Takayanagai Foundation (高柳健次郎財団) for their generous support to his research.

References

[1] T.-D. Chuang, et al., "A 59.5mW scalable/multi-view video decoder chip for Quad/3D full HDTV and video streaming applications," ISSCC 2010.

[2] D. Zhou, et al., "A 2Gpixel/s H.264/AVC HP/MVC video decoder chip for Super Hi-Vision and 3DTV/FTV applications," ISSCC 2012.

[3] C.-T. Huang, et al., “A 249Mpixel/s HEVC video-decoder chip for Quad Full HD applications,” ISSCC 2013.
[4] C.-H. Tsai, et al., “A 446.6K-gates 0.55-1.2V H.265/HEVC decoder for next generation video applications,” A-SSCC 2013.
[5] C.-C. Ju, et al., “A 0.2nJ/pixel 4K 60fps Main-10 HEVC decoder with multi-format capabilities for UHD-TV applications,” ESSCIRC 2014.
[6] D. Zhou, et al., “A 4Gpixel/s 8/10b H.265/HEVC video decoder chip for 8K Ultra HD applications,” ISSCC 2016.