A Fully-Integrated Energy-Efficient H.265/HEVC Decoder with eDRAM for Wearable Devices

Mehul Tikekar, Vivienne Sze, Anantha Chandrakasan
Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Abstract: Data movement to and from off-chip memory dominates energy consumption in most video decoders, with DRAM accesses consuming 2.8x-6x more energy than the processing itself. We present a H.265/HEVC video decoder with embedded DRAM (eDRAM) as main memory. We propose the following techniques to optimize data movement and reduce the power consumption of eDRAM: 1) lossless compression is used to store reference frames in 2x fewer eDRAM banks, reducing refresh power by 33%; 2) eDRAM banks are powered up on-demand to further reduce refresh power by 33%; 3) syntax elements are distributed to four decoder cores in a partially compressed form to reduce decoupling buffer power by 4x. These approaches reduce eDRAM power by 2x in a fully-integrated H.265/HEVC decoder with the lowest reported system power. The decoder chip requires no external components and consumes 24.9 – 30.6mW for 1920x1080 video at 24 – 50 fps.

Keywords: H.265/HEVC, Video Coding, eDRAM, Reference Frame Compression

Introduction: Wearable devices such as smartwatches, smartglasses and fitness trackers have stringent budgets for power (< 100mW) and form factor. Previous work [1-2, 5-7] has focused on video specifications of 4K and beyond which is better suited for devices with larger power budgets like smartphones, tablets, set-top boxes, etc. The large frames need to be stored in DRAM, which dominates the overall system power. For example, [1,2,7] use DDR3 memory which has a background power of 92mW [9] for the smallest chip. eDRAM helps reduce system power and physical footprint but requires *more frequent refreshes* compared to DRAM. In this paper, we propose techniques to effectively use eDRAM in a fully-integrated H.265/HEVC [3] decoder achieving 2x energy saving in eDRAM itself.

System Architecture: Fig. 1(a) shows the system block diagram of the H.265/HEVC decoder. 10.5MB eDRAM (21 x 0.5MB banks) form the main memory of the decoder. Of these, 18 banks are used for the reference frame buffer. A 3-way set-associative, 2x parallel cache of size 49kB is used to reduce the frame buffer bandwidth by 2.1x. The decoding process is split into 2 clock domains: 1) entropy decoder (CABAC) for bit-level decoding; 2) backend for pixel-level computation. The backend is parallelized over 4 cores (Dec Core 1-4). **Fig. 1(b)** shows the internal architecture of a Dec Core. Each Dec Core processes a row of Coding Tree Units (CTU: 16x16 – 64x64 pixels) in the frame as shown in **Fig. 1(c)**. Last-line dependencies between the Dec Cores are synchronized through FIFOs and access to the frame buffer is controlled through a memory arbiter.

Reference Frame Compression (RFC): eDRAM bit cells require frequent refresh to retain data, consuming 40% of total decoder power. Lightweight lossless compression is applied to reference frames to store them in 1.2x-5x fewer eDRAM banks (2x on average). The unused banks are powered down to reduce refresh power by 50%. To avoid excessive read overhead, the lossless compression is applied to small 4x4 blocks of pixels. Each 4x4 block is compressed to three elements:

1) M: minimum of the 16 pixel values [8-bit: 0 to 255]
2) R: (log-range) number of bits required to represent the delta above the minimum [4-bit: 0 to 8]
3) D: delta above M for 16 pixels using R bits (16R bits)

Fig. 2(a) shows an example of the proposed RFC algorithm. Compression is achieved since the pixels in a 4x4 block are typically correlated and R is around 3 to 4 (as opposed to 8 for uncompressed data). RFC has <1% power and area overhead but saves total power by 16%.

Unlike [8] which does not modify addressing since it uses compression to reduce the bandwidth and not storage size, we must closely pack the data to maximize the number of banks that can be powered down. The size of a compressed 4x4 block varies depending on pixel correlation, so we need an address buffer to store the mapping between 4x4 block position in the image and its address in eDRAM. To reduce the size of this address buffer, 24 consecutive addresses are packed in a 128-bit eDRAM word by storing one starting address and 24 offsets with the log-range (R) values acting as offsets. This method, shown in **Fig. 3**, reduces address buffer size by 5x (20% of total eDRAM size).

On-demand eDRAM power-up: Due to data-dependent compression of RFC, the total number of eDRAM banks needed for a frame cannot be known a priori. In a simple scheme, the maximum number of banks are powered up at the start of decoding a new frame. When decoding is complete, unused banks are powered down. **Fig. 2(b)** shows the number of eDRAM banks powered up over time.

To further reduce the number of banks, we propose an on-demand power-up scheme. At the start of decoding a new frame, one eDRAM banks is powered up for writing. When the storage utilization of the bank reaches a predetermined threshold, a new bank is powered up. The threshold is designed to take into account eDRAM startup time. This on-demand scheme for powering up banks reduces eDRAM refresh power by 33% over the simple scheme and 55% over keeping all banks powered up always.

Decoupling Buffer with Partially Compressed Data: The bit-level throughput of CABAC (Fig. 1(a)) varies widely due to varying levels of quantization. Accordingly, CABAC [4] is decoupled from the Dec Cores, which have a more regular pixel throughput. Decoupling is achieved by clocking the CABAC at a higher frequency (configurable) than the Dec Cores and adding a decoupling buffer between CABAC and Dec Cores to average out the workload variation. The data in the decoupling buffer is stored as partially decoded binary symbols (bins) rather than fully decoded syntax elements (e.g. coefficients, prediction modes, motion vectors). This reduces access to the decoupling buffer by 66x and its power by 4x. Debinarizers are needed in each Dec Core to decode the bins to syntax elements, which add an overhead of 1mW (4%).

Measurements: The test chip (**Fig. 5(a)**) can operate from 0.8V to 1.1V (eDRAM fixed at 1.1V) with an entropy decoding frequency ranging from 30.3MHz to 76.9MHz and a backend decoder frequency ranging from 7.6MHz to 19.2MHz. At maximum frequency, the chip can decode 1920x1080 video at 24 – 50 frames per second depending on encoding parameters. The chip consumes 30.6mW (0.35nJ/ pixel) for I frames, and 24.9mW (0.77nJ/pixel) for P frames.

Summary: Fig. 5(c) summarizes the specification of the H.265/HEVC decoder test chip. The test chip is implemented in 40nm CMOS. **Fig. 6** shows a comparison of the chip with the state-of-the-art video decoders [1-2, 5-7]. Energy numbers are reported at maximum throughput achieved by each chip. When scaled down to 1080p resolution, which is more suitable for wearable devices, this work achieves at least 3x lower system power. We optimize data movement using caching, RFC, and compressed data sharing between the CABAC and decoder cores. Caching reduces the frame buffer bandwidth by 2.1x. RFC reduces the number of active banks in the frame buffer by 2x and storing partially decompressed data in decoupling buffer reduces its power by 4x. Together, 2x power reduction in eDRAM or 25% lower overall system power is achieved.

Acknowledgements: TSMC University Shuttle Program provided chip fabrication and NSF provided funding.

References: [1] C.-T. Huang, ISSCC, 2013, pp.162-163. [2] D. Zhou, ISSCC, 2012, pp.224-226. [3] ITU-T Recommendation H.265: High efficiency video coding, April 2013. [4] Y.-H. Chen, TCSVT, May 2015, pp.856-868. [5] C.-H. Tsai, A-SSCC, 2013, pp.305-308. [6] C.-C. Ju, ESSCIRC, 2014, pp.195-198. [7] D. Zhou, ISSCC, 2016, pp.266-268. [8] D. Zhou, ICIP, 2014, pp.2120-2124. [9] "DDR3 SDRAM System-Power Calculator", Micron

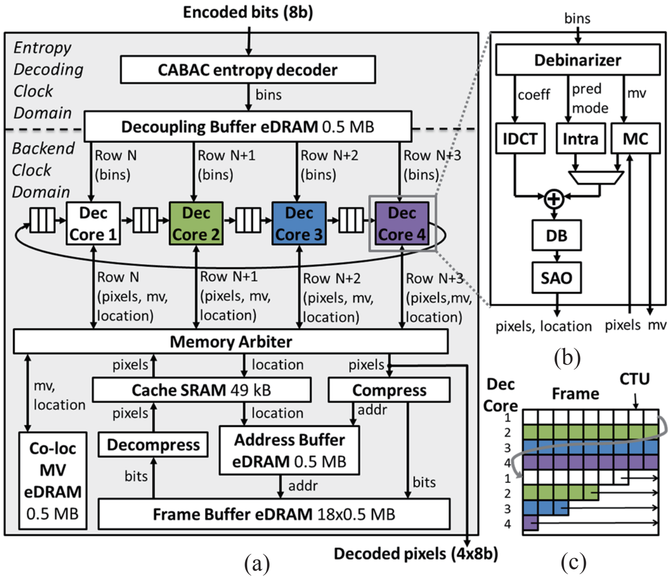
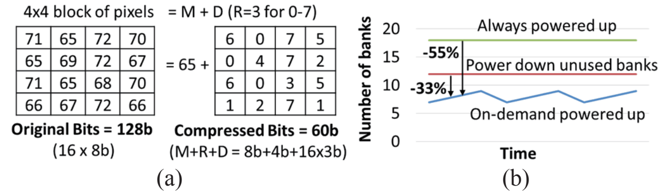


Fig. 1: (a) System architecture showing frame buffer, entropy decoder, four decoder cores, cache, compression (b) Internals of a decoder core (c) Processing order of 4 decoder cores in a picture



4x4 block of pixels = M + D (R=3 for 0-7)

71	65	72	70
65	69	72	67
71	65	68	70
66	67	72	66

= 65 +

6	0	7	5
0	4	7	2
6	0	3	5
1	2	7	1

Original Bits = 128b (16 x 8b)

Compressed Bits = 60b (M+R+D = 8b+4b+16x3b)

Fig. 2: (a) Example of Reference Frame Compression (RFC) algorithm (b) On-demand power-up of eDRAM banks saves 33% refresh power compared to powering down unused banks and 55% compared to always powered-up case

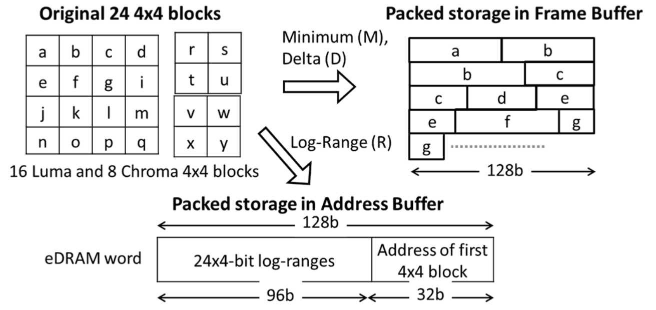


Fig. 3: Packed storage formats save 2x Frame Buffer size and 5x Address Buffer size

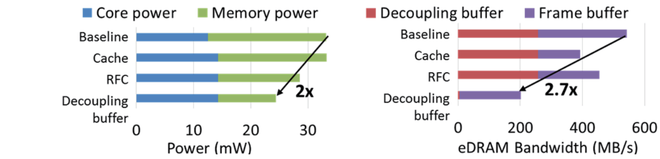
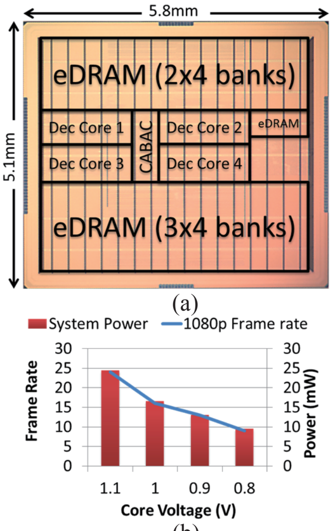


Fig. 4: Proposed optimizations reduce eDRAM power by 2x and bandwidth by 2.7x



Technology	TSMC 40nm LP
Supply Voltage	Core 0.8 - 1.1V, eDRAM 1.1V, I/O 2.5V
Standard	H.265/HEVC (Main Profile with 2 reference frames)
Chip Size	5.8 mm x 5.1 mm
Core Size	5.1 mm x 4.3 mm
Gate Count	1,438 kgates (NAND2)
On-Chip SRAM	162.75 kB
On-Chip eDRAM	21 x 0.5MB
Max Resolution	1920 x 1080
Max Throughput	47.9Mpixels/s
Power at 1.1V	30.6mW (I Frame) 24.9mW (P Frame)

(c)

Fig. 5: (a) Die micrograph (b) Measured core voltage scaling for power and performance (1080p P frames) (c) Summary of chip specifications

	This Work	ISSCC 2013 [1]	A-SSCC 2013 [5]	ESSCRIC 2014 [6]	ISSSCC 2016 [7]	ISSCC 2012 [2]
Standard	H.265/ HEVC	H.265/ HEVC WD4	H.265/ HEVC	H.265/HEVC multistandard	H.265/ HEVC	H.264/AVC MP/MVC
Gate Count	1438K	715K	446K	3454K	2887K	1338K
SRAM	162.75kB	124kB	10.2kB	154kB	396kB	79.9kB
Technology	40nm/1.1V	40nm/0.9V	90nm/1V	28nm/0.9V	40nm/1V	65nm/1.2V
Frame buffer Storage	128b eDRAM	32b DDR3	n/a	32b LPDDR3	64b DDR3	64b DDR3
Max Throughput	1920x1080 @24fps	3840x2160 @30fps	1920x1080 @35fps	3840x2160 @60fps	7640x4320 @120fps	7640x4320 @60fps
Core Power* [mW]	21.2 [I] 14.6 [P]	76	36.9	104	690	410
Frame buffer Power* [mW]	9.4 [I] 10.3 [P]	219	n/a	n/a	1198	2520
Core energy* [nJ/pixel]	0.25 [I] 0.45 [P]	0.31	0.59	0.20	0.15 - 0.25	0.21
Frame buffer* energy [nJ/pixel]	0.11 [I] 0.32 [P]	0.88	n/a	n/a	0.30	1.27
System energy* [nJ/pixel]	0.35 [I] 0.77 [P]	1.19	n/a	n/a	0.45 – 0.55	1.48
System power at 1080p [mW]	24.9 mW – 30.6 mW	> 92mW	n/a	n/a	> 92mW	> 92mW

Fig. 6: Comparison with state-of-the-art video decoders. (I = I frames, P = P frames. * = power at max throughput. Measured results in this work were averaged across four sequences: Ki-mono, BQTerrance, Cactus, ParkScene.)