IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 31, 2022, accepted February 18, 2022, date of publication March 4, 2022, date of current version March 14, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3156274

Content-Adaptable ROI-Aware Video Storage for Power-Quality Scalable Mobile Streaming

ALI HAIDOUS[1], (Member, IEEE), WILLIAM OSWALD[2], HRITOM DAS[2], (Member, IEEE), AND NA GONG[2], (Member, IEEE)

[1]Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND 58102, USA
[2]Department of Electrical and Computer Engineering, University of South Alabama, Mobile, AL 36688, USA

Corresponding author: Na Gong (nagong@southalabama.edu)

This work was supported by the National Science Foundation under Grant CCF-1855706.

ABSTRACT The demand for mobile video streams is constantly increasing. With this demand comes a need for mobile devices to receive more videos at ever increasing quality. However, due to the large size of video data and intensive computational requirements, video streaming requires frequent memory access that consume a substantial amount of mobile device power; as a result, the battery life of mobile devices is limited. In this paper, we present a video content-adaptable Region-of-Interest (ROI)-aware video storage technique that promotes power savings. During the video encoding process on the transmitting server, based on the macroblock variance and ROI characterization, the ''macroblocks of interest'' are identified and embedded in the encoded bitstream. In the decoding process, a new frame buffer with dynamic power-quality trade-off is presented to adapt to the macroblock characteristics during run-time. Results from the system-level and circuit-level simulations show that the proposed technique enables substantially more truncated bits and significant power savings while delivering similar or better video quality as compared to other state-of-the-art solutions.

INDEX TERMS Content, memory, region of interest, power consumption, video quality.

I. INTRODUCTION

Recently, mobile video streaming on YouTube, Vimeo, and Netflix has increased quickly and will consume approximately 79% of the total internet traffic by 2027 [1]. At the same time, power-efficient video storage has proven to be a very challenging problem to solve. This is due to the large data sizes associated and intensive computational requirements demanding frequent data access. With the advancement of computing technologies, more video streaming services deliver content to battery-powered mobile devices: such as smart phones and Internet-of-Things (IoT). On one hand, these devices would benefit greatly from low-power consumption as this would extend their battery life. On the other hand, the mobile video streaming process – receive, decode, and display of a video bitstream – consumes considerable power and limits the mobile devices' battery life. For example, with a video decoding chip, embedded memories contribute to over 50% of the decoding power consumption [2]. This use-case is only expected to grow for the next-generation video formats, H.265/HEVC and H.266/VVC, which has 2x-3x greater memory demands when compared to H.264 [3].

Today's mobile hardware designers, including memory designers, are focusing on hardware-level energy-efficient design techniques in order to accommodate the large amount of video data. However, these design techniques usually come with significant implementation overhead (e.g., silicon area, delay) to solve failure problems in memories. We have recently explored viewer-aware video memory design by investigating the impact of illuminance levels in different viewing surroundings on the viewer's experience [4]–[7], as shown in Fig. 1. Our previous studies illustrate a new dimension of power savings for hardware design through the introduction of viewer awareness, but the developed memories cannot adapt to a wide variety of mobile videos. To enable an optimized trade-off between power efficiency and video quality, this paper aims to develop a video content-adaptable Region-of-Interest (ROI)-aware memory for general videos. Specifically, this paper makes the following contributions:

- An intelligent ROI-aware and content-adaptive framework is proposed to determine video frame regions

The associate editor coordinating the review of this manuscript and approving it for publication was Wu-Shiung Feng.

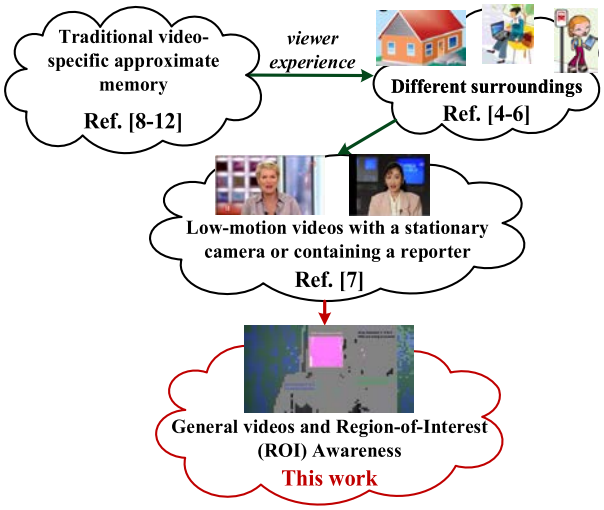


FIGURE 1. Proposed content-adaptable ROI-aware low-power video memory. The direction of an arrow indicates the development from one previous technique to a newer one.

to preserve (output quality) or truncate bits for power savings. The truncation is applied for all Luma and Chroma video data (i.e., Y, U, and/or V components) (Section III).

- The system-level implementation scheme of the proposed technique is developed and discussed (Sections IV-A, IV-B, and IV-C).
- A low-power low-cost frame buffer with dynamic power-quality trade-off is developed to adapt to the video content (i.e., macroblock characteristics) during run-time (Section IV-D).
- A comprehensive suite of simulations on the proposed technique is performed and the enriched results are discussed, including the performance, circuit-level power efficiency, video-level power efficiency, number of truncated bits, and output quality of various mobile videos (Sections VI-A, VI-B, VI-C, and VI-D).
- An extensive statistical analysis demonstrates the effectiveness of the proposed technique in achieving significant bit truncations and power savings as compared to the state-of-the art, particularly for the videos with medium or high variance (Section VI-E).

To the best of the authors' knowledge, this is the first work that seamlessly integrates ROI knowledge, i.e., ''macroblocks of interest'', into the hardware design process.

The organization of the paper is as follows. A review of low-power video memory designs is provided in Section II, Section III presents the macroblock variance and ROI study, and Section IV discusses the proposed technique. We discuss the evaluation methodology and results in Sections V and VI respectively, and finally, we conclude the paper in Section VII.

II. STATE OF THE ART

A vast amount of research has been conducted to improve the power efficiency of video data storage. State-of-the-art, power-efficient video memories consist of either approximate memory with application-level information [8]–[12] or viewer-aware memories with an awareness of viewer's experience [4]–[7]. In this section, some of the existing work related to the proposed technique are briefly reviewed, and the detailed comparison analysis will be provided in Section VII.

A. APPROXIMATE VIDEO-SPECIFIC MEMORY

Researchers have presented various low-power video memory design techniques. Chang *et al.* [8] presented a hybrid 6T + 8T SRAM to achieve quality-power optimization. Gong *et al.* [9] developed a hybrid 8T + 10T memory for power savings based on the correlation between most-significant-bits (MSBs) of video data. In [10], a heterogeneous sizing scheme was presented to reduce the failure probability of conventional 6T bitcells. The video memory presented in [11] used the Least-Significant-Bits (LSBs) of video data to store the MSBs' error-correction-code (ECC). Kazimirsky *et al.* [12] developed a hybrid SRAM + DRAM memory to store MSBs in robust SRAM bitcells and LSBs in error-prone DRAM bitcells, leading to a tolerable output quality with power reduction. However, all of those video memory designs were developed without considering viewer's experience.

B. VIEWER-AWARE VIDEO MEMORY

We have investigated viewer-aware low-power video memory techniques in [4]–[6]: where an increased amount of ambient luminance allows for a larger number of bits to be truncated without noticeable degradation to the viewers. Very recently, we studied the impact of video content characteristics on viewer's experience to enable video content-adaptive memory with dynamic energy-quality tradeoff [7]. However, the technique determined the number of truncated LSBs based on the averaged plain macroblock percentage of an entire video sample; therefore, it was only effective to store low-motion videos with a stationary camera or containing a reporter in a video cast use-case. Additionally, this technique may result in noticeable distortion, e.g., a banding distortion caused by bit truncation, which negatively influenced the viewer's experience.

The common feature of these viewer-aware storage techniques is that the same number of the truncated bits were applied on an entire video. In contrast, the technique proposed in this paper realizes content adaptation and ROI awareness within each video frame, thereby maximizing the number of truncated bits while maintaining the video quality.

III. OVERVIEW OF THE PROPOSED TECHNIQUE

In this section, we first present the motivation of the proposed technique that introduces ROI awareness as bit truncation is applied for power savings. Then, the high-level overview of the proposed technique is shown.

A. MOTIVATIONAL EXAMPLE

Researchers conducted studies on the human visual system's (HVS) performance and concluded that viewers usually

(a) Output quality using [7] (at 3 truncated bits)

(b) Output quality of the proposed technique (at 3 truncated bits)

(c) [7] (left) vs. Proposed technique (right)

FIGURE 2. Observer discernable flaws in the facial region due to a "banding effect" on the face when comparing (a) and (b) caused the overall quality of the frame to become unacceptable at 3 truncated bits (Video tag: wF6lvdXXwc4 from [14]).

pay more attention to one or a few areas of a video and the region of concentration is called Region-Of-Interest (ROI) [13]. For example, in video conferencing applications, viewers typically pay more attention to the face regions than other areas. In video surveillance, the facial regions are what viewers concentrate most on in consecutive frames. Accordingly, ROIs have higher contribution towards the overall visual quality than other areas. Consequently, if truncation-caused banding distortion appears in ROIs, this will negatively influence a viewer's experience. Fig. 2 shows one example. The output quality of the video (Video tag: wF6lvdXXwc4 [14]) using the technique in [7] is shown in Fig. 2 (a). Since the banding distortion caused by bit truncation appears on the reporter's face, viewers were

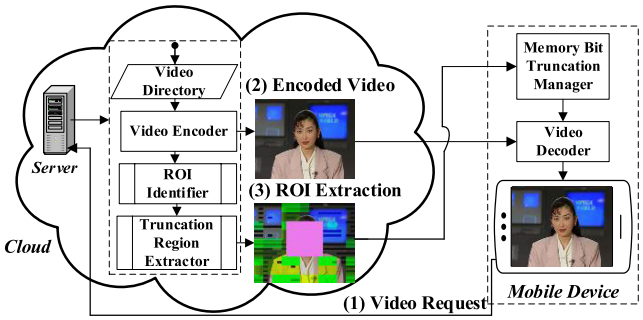


FIGURE 3. Proposed region-of-interest and macroblock texture framework.

less likely to accept the displayed degradation due to this particularly noticeable distortion, as emphasized in [7].

Therefore, the motivation for this work arises from the following two observations:

1) In a video frame, the distortion in ROIs is more noticeable by viewers. Accordingly, if ROIs can be extracted and protected from truncation, the video quality would be improved from the viewer's perspective (Fig. 2 (b)). A comparison of the report's face using the technique in [7] and the proposed technique with ROI awareness is shown in Fig. 2 (c).

2) There existed a positive correlation between power savings and the number of bits truncated in a video decoder's frame buffer memory [7]. To optimize the power efficiency, it would be beneficial to increase the number of truncated bits in other regions which are not ROIs: the truncation regions.

B. OVERVIEW OF THE PROPOSED CONTENT-ADAPTABLE ROI-AWARE VIDEO STORAGE

Fig. 3 shows the proposed content-adaptable ROI-aware video storage technique. During the traditional mobile video streaming process, first, from (1) in Fig. 3, the mobile device requests a video for display from the cloud. Then, the streaming servers process the requested video by encoding and transmitting the encoded bitstream to the mobile device for decoding and display, (2) in Fig. 3. During this process, multiple memories are needed for storing the intermediate and final results of the frame data. In particular, the reference macroblock, frame memory, and display memory, which store the decoded video frames, are accessed very frequently, and they have a profound impact on the system's overall cost and power consumption. The proposed technique extracts ROIs in the cloud server and transmits the truncation region data together with the encoded bitstream to the mobile device, (3) in Fig. 3, to further reduce the mobile device's power consumption from computational overhead. The mobile device hardware video decoder receives the truncation region data and makes memory bit-truncation decisions for greater power savings with less perceived quality loss than [7]. To optimize the truncation decision logic of the mobile device hardware, which further improves power consumption, either *no truncation* or *3-bit truncation* is applied to the truncation

regions. Explicitly, the proposed technique is detailed as follows.

1) ROI AWARENESS

ROI has been recently applied for different research areas for video system optimization, such as wireless transmission [15], virtual reality (VR) [16], and video summarization [17]. The proposed technique introduces ROI awareness into video storage. Specifically, to minimize the complexity and computational overhead, we focus on the faces as ROIs in our analysis based on the basic machine learning facial detection OpenCV model [18]. Different algorithms, such as user attention model [13], motion-based models [17], and machine learning models [19], can be applied in our future investigations to extract different ROIs. It should be noted that the complexity of ROI extraction algorithms is a trade-off between video quality, computational complexity, and power consumption. A simple ROI extraction algorithm will save compute resources and power from video encoding. Also, the algorithm may transmit fewer truncation region bits to mobile devices, thus more pixel bits will be truncated for power savings in the mobile devices. The drawback is that it will influence the video quality negatively. Alternatively, a more complex ROI algorithm will identify additional regions and therefore, can convert a video without ROI to a video with ROI, which will benefit the video quality, but it will reduce the power savings due to the less truncated bits and increased computational complexity.

2) VIDEO CONTENT ADAPTATION

After the ROIs to preserve are detected and captured by the framework ROI Identifier, it then searches for regions of low variance measured by the percentage of plain macroblocks (MBs). Specifically, a MB defines an area of 16 × 16 pixels within a frame. An attribute associated with MBs is how ''Textured or Plain'' they are. A Plain MB is one in which the variance of intensity within the MB is less than or equal to the threshold value. It has been concluded in [7] that textured MBs are less susceptible to bit-truncation. We will adopt the pre-established method from [20] for determining the variance in a MB.

$$V_{MB} = \sum_{i=0}^{15} \sum_{j=0}^{15} (P(i,j) - \rho_{MB})^2 \gg 8 \quad (1)$$

$$\text{MB} = \begin{cases} \text{Plain}, & \text{if}(V_{MB} \leq \text{Th}_{low}) \\ \text{Textured}, & \text{Else} \end{cases} \quad (2)$$

Equations (1) and (2), where ρ_{MB} is the average brightness within the MB, V_{MB} is the texture variance within the MB, and traditionally, Th_{low} is defined as a value of 1.25 [21].

3) TRUNCATION REGION EXTRACTOR

After ROIs are identified on the server, a truncation region extractor encodes the truncation region data using a proprietary protocol per frame and transmits in synchronization with the encoded video transmission to the mobile device. The truncation region data is decoded onboard the mobile device's hardware video decoder in a novel Memory Bit Truncation Manager (MBTM) hardware unit: which truncates a novel frame buffer memory through the use of unique control YUV truncation signals. The video decoding and bit truncation processes occur in lockstep.

4) 3-BIT TRUNCATION

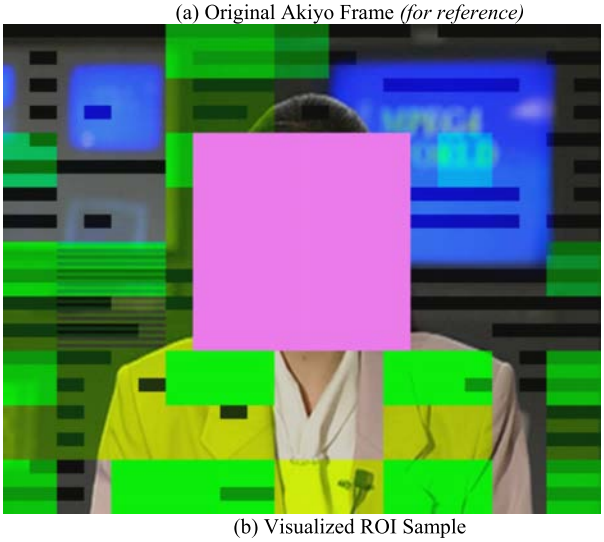
Truncation is performed in the YUV (Y'CbCr) color space [22], inferring that any truncation is done to the YUV color values. The memory designed in [7] truncated 1, 2, or 3 bits in the Least Significant Bits (LSBs) of the Y vector[1] of all frames within an entire video as a blanket truncation. The proposed technique will enable a different amount of truncated bits for each region within each frame within an entire video. To minimize the implementation overhead, only 3-bit truncation is adopted in the new frame buffer, which will be discussed in Section IV-D. Meanwhile, the proposed technique can identify bit-truncation for each Y, U, and V vector of the frame separately for each truncation region in each frame, instead of only truncating the Y vector as a blanket truncation across the entire video as the existing techniques [4], [5], [7]. Furthermore, the proposed technique is expected to enable additional bit truncations as compared to existing techniques. Also, to minimize the video quality degradation caused by bit truncation, the developed frame buffer truncates three LSBs to the optimal value ''100'' [7], instead of truncating the values to ''000''.

Fig. 4 shows the *Akiyo* video sample using the proposed technique. The extracted preserved ROI region is highlighted in pink. All truncation regions within a frame are identified, including the following seven possible truncation combinations: (1) Green, Y vector truncation; (2) Blue, U vector truncation; (3) Yellow, V vector truncation; (3) Dark blue, YU vectors truncation; (5) Dark Yellow, UV vector truncation; (6) Dark green, YV vectors truncation; and (7) Grey, YUV vectors truncation. Each of these combinations would be encoded in the truncation region data for the MBTM to generate control signals for memory bit truncation in the video decoding process.

To conclude, our proposed technique truncates the chroma sub samples within each frame as well as the luminosity: Y, U, and V vectors. Previous research only targeted luminosity, Y, of a video for truncation, while chroma samples were disregarded for the entire video. Also, our technique preserves ROIs that impact viewer perception most, while enabling greater truncation for each Y, U, and V vector for the truncation regions with textured MBs. Accordingly, the proposed technique will realize a greater number of truncation while preserving visual quality. The system-level and circuit-level implementations of the proposed technique will be discussed in Section IV.

[1]The word *vector* means all bytes of a component – Y, U, or V – in one frame.

(a) Original Akiyo Frame *(for reference)*



(b) Visualized ROI Sample

FIGURE 4. Akiyo from [28], sample visualized, as generated internal to the proposed method's frame parsing process. Pink, preserved ROI. Seven possible truncation combinations: 1. Green, Y vector truncation. 2. Blue, U vector truncation. 3. Yellow, V vector truncation. 4. Dark blue, YU vectors truncation. 5. Dark Yellow, UV vector truncation. 6. Dark green, YV vectors truncation. 7. Grey, YUV vectors truncation.

IV. PROPOSED TECHNIQUE: SYSTEM LEVEL AND CIRCUIT LEVEL IMPLEMENTATION

This section presents the system-level and circuit-level implementation of the proposed technique.

A. SYSTEM-LEVEL IMPLEMENTATION: VIDEO STREAMING PLATFORM

Fig. 5 shows the developed system-level video streaming platform. As shown, a Raspberry Pi [23] microcontroller was used to serve as a video streaming server with which a mobile device would communicate and retrieve video data. Also, we utilized a Z-Turn 7020 [24] board and synthesized an H.264 video decoder into the on-board Xilinx Zynq 7020 Field Programmable Gate Array (FPGA) which would operate as a mobile device. Finally, we captured the decoded video data via a Magewell [25] HDMI Video Capture & Display Device.

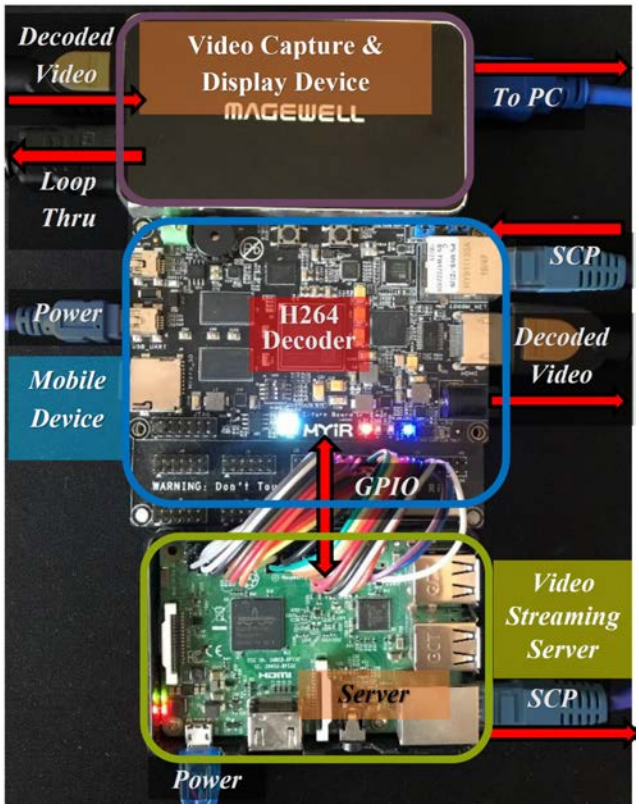


FIGURE 5. H264 video stream demonstration platform hardware system.

The corresponding block diagram for Fig. 5 is illustrated in Fig. 6. The video streaming process is kicked-off by a command from the mobile device to the server to retrieve an encoded H.264 video stream over Secure Copy Protocol (SCP) [26]. The mobile device sends the initial kick-off command to the server over a serial terminal on a PC interfaced with the mobile device over USB. The server then processes the video stream requested by the mobile device by both transmitting an H.264 encoded format of the video stream over SCP to the mobile device and parsing the frames for truncation region information.

After the video frame is parsed on the server, the truncation region information is transmitted over GPIO per frame. In our developed system, the protocol is defined in Table 1. Only the truncation region information of the frames that would be truncated is transmitted. The preserved ROI information will not be transmitted as these regions are identified prior to the transmission on the server and preserved. As listed in Table 1, the first index, index 0, denotes the current frame number parsed. The second index, index 1, denotes the number of truncation regions to truncate. Then the next indices denote the first three YUV truncation signal bits plus two sets of XY coordinates denoting the left top and right bottom corners of rectangles grouping the affected truncation region. These three indices repeat for each region called out by the ''Number of Regions'', index 1. The GPIO interface data width bit size of the developed system is 22-bits per index. The 22-bit distribution is to account for a

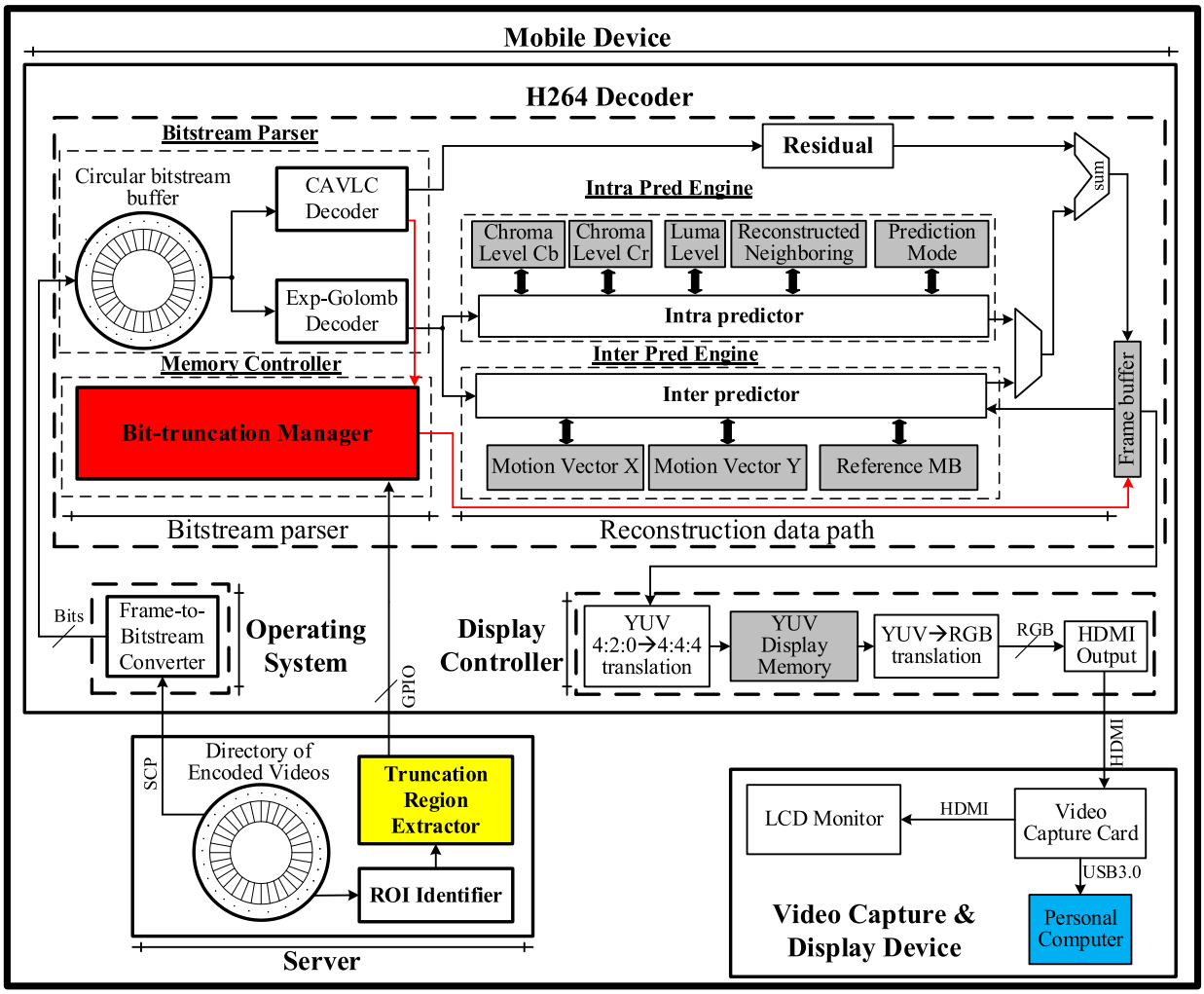


FIGURE 6. Mobile video steaming system block diagram.

TABLE 1. Truncation region GPIO protocol. (a) Server-to-mobile device. (b) Mobile device-to-server.

(a) SERVER-TO-MOBILE DEVICE

Index 0	Index 1	Index 2	Index 3	Index 4	…	Index N+1	Index N+2	Index N+3
Frame Number	Number of Regions	YUV_1 Truncation	(X_I1,Y_I1)	(X_I2,Y_I2)	…	YUV_N Truncation	(X_N1,Y_N1)	(X_N2,Y_N2)
22 bits	16 bits	3 bits	22 bits	22 bits	…	3 bits	22 bits	22 bits

(b) MOBILE DEVICE-TO-SERVER

Index 0	Index 1
Frame Number Request	Send Frame Flag
22 bits	1 bit

maximum of $2^{11} \times 2^{11}$ pixel addressing – a max resolution of 1,920 × 1,080 – totaling 22 bits. There is an additional 2 handshaking bits between the server and mobile device to denote data reception confirmation in-order to transmit the next index.

This truncation region information will be transmitted to a MBTM for processing in the mobile device side, as discussed in Section IV-B. The MBTM will generate control signals for the frame buffer memory, thereby determining which sub-pixels – from Y, U, and/or V – shall be truncated for each frame written to the frame buffer memory, which will be detailed in Section IV-D. Finally, the decoded and bit-truncated frame is output over HDMI from the mobile Device and captured by the Video Capture & Display Device.

B. MEMORY BIT TRUNCATION MANAGER (MBTM)

The MBTM implemented into the H.264 decoder parses the protocol data that is transmitted by the server's Truncation Region Extractor. The flow is broken down as follows. First, from Fig. 7 (a), the encoded frame is transmitted via SCP to the mobile device. Fig. 7 (b) illustrates the truncation regions determined to be bit-truncation capable on a sub-frame vector level: Y vector, U vector, and V vector each encompassing

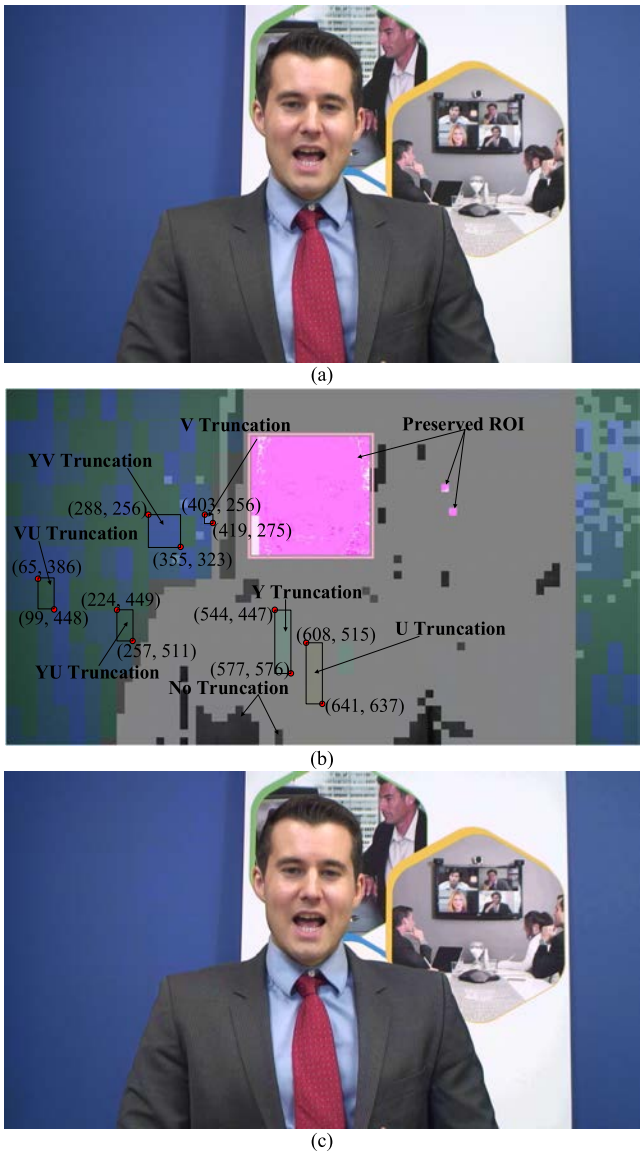


FIGURE 7. (a) Encoded frame 175 from Johnny_1280 × 720_60 video [28]. (b) Visual of areas being truncated. 45 regions total. (c) Output decoded frame. 2,282,496 bits truncated.

all the sub-frames summing to a frame. From Fig. 7 (b), the gray areas denote the truncation regions determined to be bit-truncation capable for all Y, U, and V vectors. The areas in boxes are regions where only 1 or 2 vectors were determined to be bit-truncation capable. Two coordinates, top-left and bottom-right, are highlighted in Fig. 7 (b) for each of these regions to show how the truncation region data was used to determine the regions to truncate using the protocol in Table 1. A total of 61 regions to truncate are shown in Fig. 7(b). Fig. 7 (c) shows the resultant frame after Fig. 7 (a) is decoded using the identified truncation region information. As shown, the preserved ROI around the face, pink region from (b), is not truncated to avoid visual quality degradation. The frame is decoded normally, but when it is written into the frame buffer, the transmitted truncation region information is used to control the T_Y, T_U, and T_V control inputs to truncate the frame buffer memory as it is written. These control inputs are provided to [28] in detail in Section IV-D.

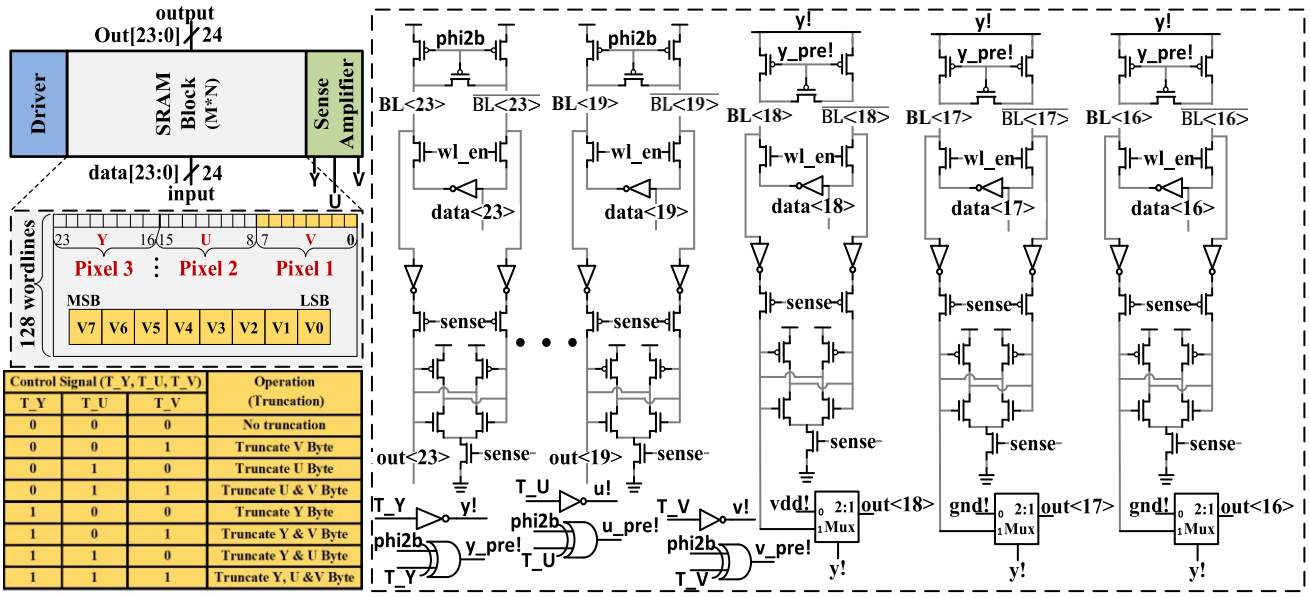
C. H.264 DECODER AND MBTM INTEGRATION

A H.264 video decoder is implemented based on the Open Source Osenlogic OSD10 decoder IP [27]. This decoder was capable of decoding baseline profile level 3.1 encoded bitstreams. The slice types supported were I-Slice, SI-Slice, P-Slice, and SP-Slice [28]. The entropy coding profile supported was Context-Adaptive Variable-Length Coding (CAVLC). The decoder took an H264 Network Abstraction Layer (NAL) bitstream and output YUV 4:2:0.

During the NAL bitstream parsing process, the bitstream is parsed into raw bytes of syntax elements from the Raw Byte Sequence Payload (RBSP). Within the RBSP, therein lies the slice layers. Ignoring the Sequence Parameter Set (SPS) and the Picture Parameter Set (PPS), the Instantaneous Decoder Refresh Access Unit (IDR Slice(s)) and the slice layer includes all slice headers and slice data for the frames that shall be truncated using the MBTM. H.264/AVC defines a frame as an array of luma samples and two corresponding arrays of chroma samples: denoted as YUV.

Specifically, the slice header includes the parameters *first_mb_in_slice*, which indicates the position of the first macroblock in the slice data, and *frame_num*, which represents the order in which a video decoder shall decode the encoded frames. This is not the same as the display order or Picture Order Count (POC), which is the order in which the frames are displayed. The *frame_num* parameter is used to determine which frames during the decoding process would be susceptible to YUV bit-truncation by the MBTM and the *first_mb_in_slice* is used to determine the starting coordinates of the macroblocks susceptible to bit-truncation. The slice data included all the macroblocks of the slice.

After the MBTM determined that a frame would be truncated, through a conditional match between the frame number parameter from Table 1 (a) and *frame_num*, a running count of the current macroblock index was kept track of internally to the MBTM from the slice data starting with the index of *first_mb_in_slice*. After the MBTM determined that a macroblock would be truncated, through a conditional match of the running macroblock index and the truncation region given by the two indices from Table 1 (a) that indicate the rectangular region which YUV truncation would be applied, the MBTM passes through the YUV truncation signal, from Table 1 (a), to the frame buffer which would result in the macroblock being truncated to the desired amount. An internal signal denoting the number of macroblocks truncated in the frame is then incremented. After all the macroblocks desired to be truncated in the frame are truncated, denoted by the number of ROI parameter from Table 1 (a), then Table 1 (b), the Send Frame Flag is set then reset by the MBTM over GPIO to signal the next frame information to truncate. From Table 1 (b), the Frame Number Request index is used to fetch any frame index truncation information for macroblocks that required multiple frames



Control Signal (T_Y, T_U, T_V)			Operation (Truncation)
T_Y	T_U	T_V	
0	0	0	No truncation
0	0	1	Truncate V Byte
0	1	0	Truncate U Byte
0	1	1	Truncate U & V Byte
1	0	0	Truncate Y Byte
1	0	1	Truncate Y & V Byte
1	1	0	Truncate Y & U Byte
1	1	1	Truncate Y, U &V Byte

FIGURE 8. Circuit-Level implementation of the proposed frame buffer memory.

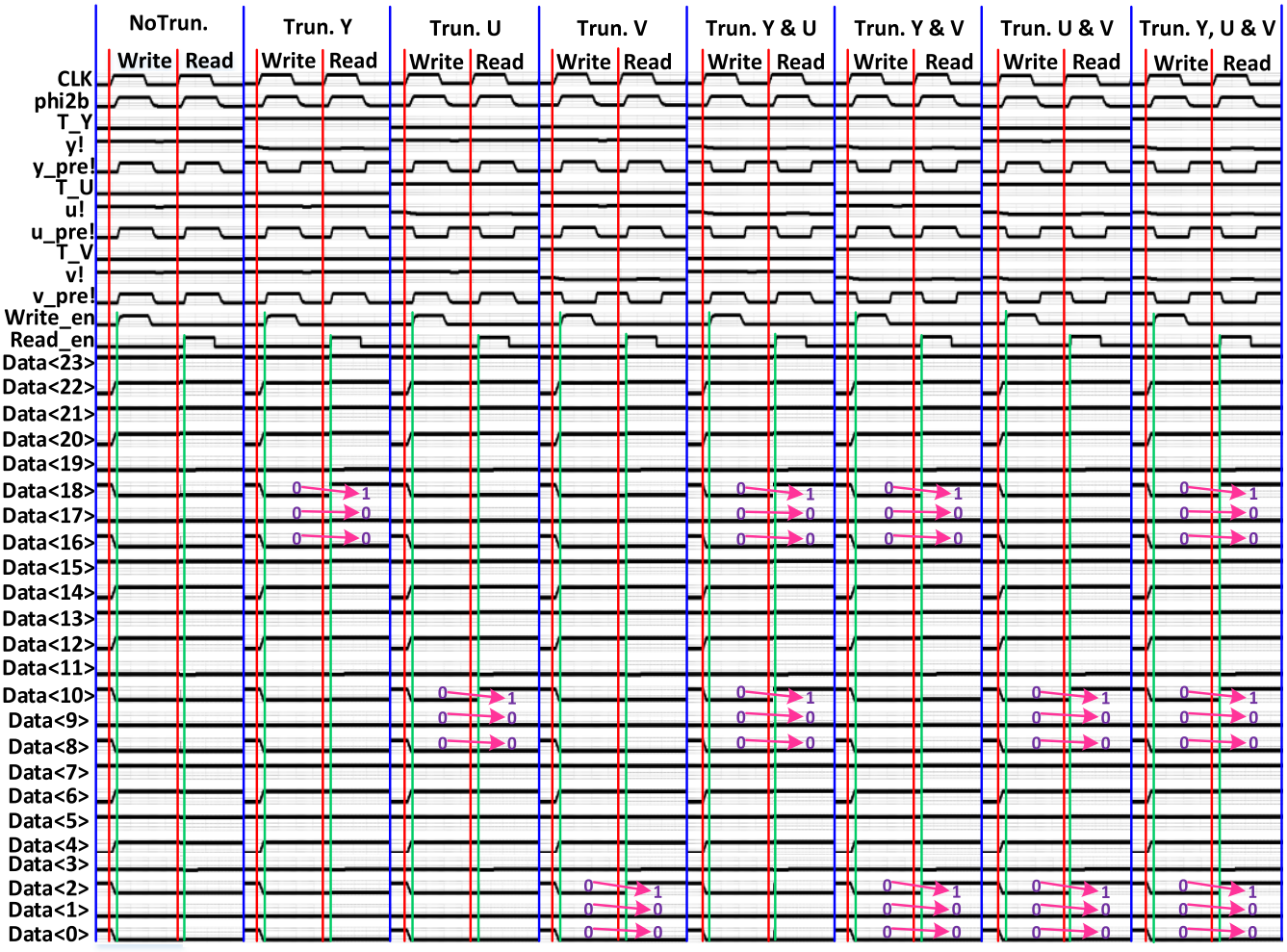


FIGURE 9. Timing diagram of the frame buffer circuit.

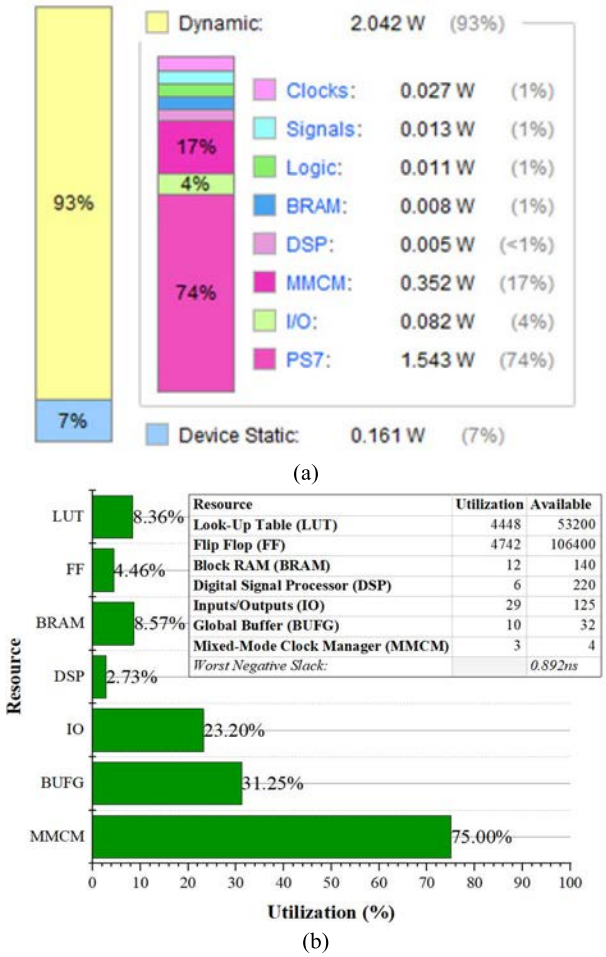


FIGURE 10. Hardware FPGA system implementation results *without* BTM. (a) On-Chip Power, Total Power: 2.203W. (b) Resource allocation.

for prediction. This process is repeated until the end of the NAL bitstream.

The trade-off with utilizing the BTM is the additional GPIO parallel bitstream overhead required to truncate the macroblocks in each frame. Each frame parsed had an absolute worst case overhead of approximately 380,738 additional bits to transmit using the protocol from Table 1. This worst case is calculated assuming every macroblock with 16 × 16 pixels in a maximum resolution of 1,920 × 1,080 would be truncated differently per frame in a video. On average, however, the number of additional bits transmitted per frame is 1,200, because the maximum resolution of each frame is 1920 × 1080 and the truncation regions are combined to encompass a greater area in the video to save on bits transmitted: on average 50 truncation regions per frame. With a 1920 × 1080 video at 30 frames per second progressive (1080p 30fps) or a 1280 × 720 videos at 60 frames per second progressive (720p 60fps), i.e. 5,000 kbps bit rate, the worst case percentage overhead would be 7.62% with an average of 0.02% per frame. The protocol utilized is one of the simplest methods to implement the proposed technique.

D. CIRCUIT-LEVEL IMPLEMENTATION OF THE PROPOSED FRAME BUFFER MEMORY

During the video decoding process, multiple memories are needed. In particular, the frame buffer memory is accessed very frequently and it has a profound influence on the system's overall cost and power consumption [7]. In this paper, a new frame buffer is designed, and the circuit-level implementation is shown in Fig. 8. Specifically, the logic in the truth table highlighted in yellow was designed to be supported by the MBTM. Here, T_Y, T_U, and T_V are utilized to truncate Y, U, and V byte from the word. Each word consists of a Y, U, and V byte. During the Write Enable (WE) phase of the frame buffer memory access, if either control line of T_Y, T_U, and / or T_V are asserted, the memory would truncate the 3-LSB of the optimal asserted vector as "100" [7].

The proposed frame buffer has M words and each word consists of N bits. To evaluate the functionality and measure average power consumption of this proposed circuitry, a 128-word by 24-bits memory array is designed. Here, input and output pins are denoted as data[23:0] and out[23:0] respectively. Bits 23-16 are named Y byte, bits 15-8 are named U byte, and bits 7-0 are named V byte. The memory implemented had a driver and sense amplifier for writing and reading data. These enabled bit truncation according to T_Y, T_U, and T_V control signal activation. If T_Y, T_ U, and T_V are all de-asserted as logic '0', then the frame buffer would operate as a traditional memory device where the sense amplifier would operate with a supply voltage (VDD) and pre-charge signal phi2b. When the T_Y signal is asserted as logic '1', the peripheral circuitry would generate two signals: y! which is the inverted value of T_Y and y_pre! which is inverted value of the pre-charge enable signal. These two signals are used to control the sense amplifier for the Y byte's 3-LSBs, thereby enabling truncation. During this process, the VDD for this sense amplifier remains grounded and the pre-charge signal would be reactivated. As a result, the power consumption of this portion of circuitry will be reduced as compared to the normal operation. During the read back operation, the 3-LSBs are generated as "100" though use of three 2:1 multiplexers in-place of regular of data output. When the bit truncation is asserted, these multiplexers would select "100" through control signals y!, u!, or v!. Otherwise, these multiplexers would pass normal readout data values. In addition, the VDD of all the 3 LSBs of each byte are also controlled by the corresponding control signals y!, u!, and v!. During the truncation, VDD for LSBs can be powered off to save power consumption and multiplexers will select "100" as the output data, thereby achieving low-power operation. The detailed timing diagram and power efficiency of the proposed memory will be discussed in Section VI.

V. EXPERIMENTAL METHODOLOGIES

This section discusses the metrics, methods, and strategies used to evaluate the effectiveness of the proposed technique.

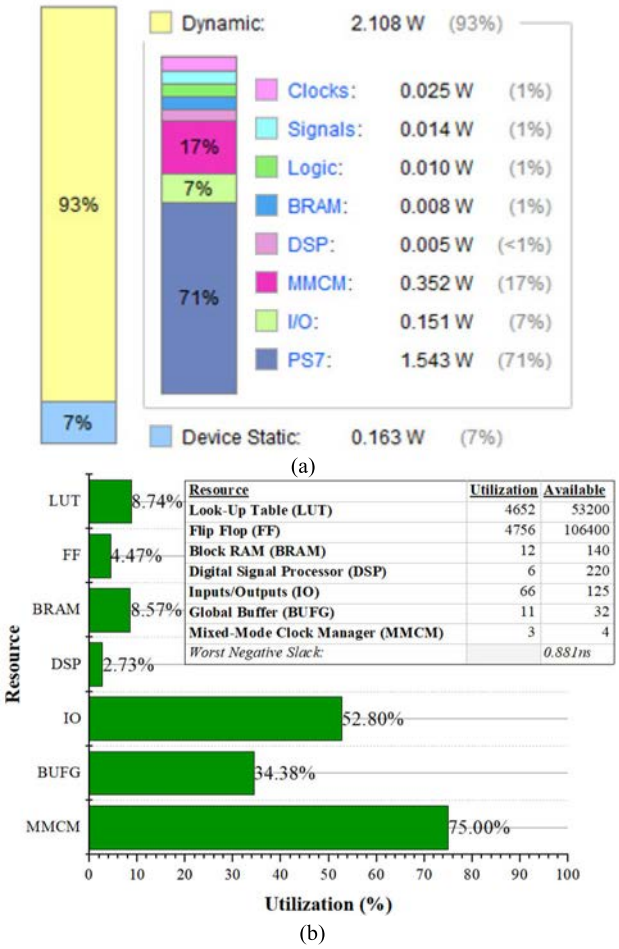


FIGURE 11. Hardware FPGA system implementation results *with* BTM. (a) On-Chip Power, Total Power: 2.271W. (b) Resource allocation.

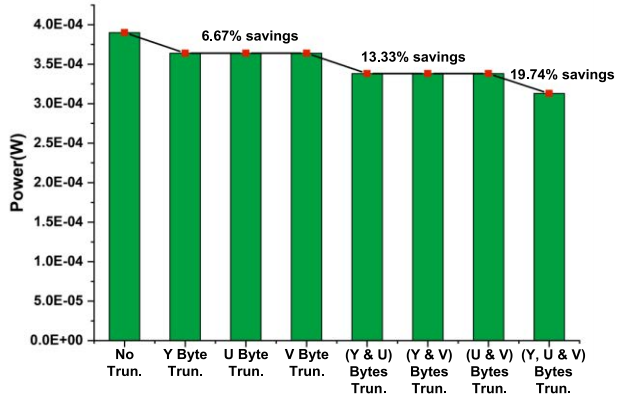


FIGURE 12. Power savings (one word) of the frame buffer circuit.

The testing and analysis setup used to generate the experimental results is also discussed.

A. VIDEO SELECTION

To verify the effectiveness of the proposed technique, such as video quality, power savings, and number of truncated bits, 74 videos with diverse characteristics were selected from Xiph.org Video Test Media [29]. Of those videos, 60 videos contain facial features to enable ROI preservation using the proposed technique. All videos were converted to the YUV 4:2:0 chroma subsampling standard for ease of bit-truncation. The results are presented in Sections V-B to D. We further conduct statistical analysis using all 889 videos of 1080p resolution or lower selected from the YouTube UGC dataset [30] and Xiph.org Video Test Media [29]. Of those 889 videos, 699 videos contain facial features for ROI analysis, which will be detailed in Section VI-E.

B. VIDEO FRAME QUALITY METRICS

Existing video quality metrics such as Peak Signal-to-Noise Ratio (PSNR) and structural similarity (SSIM) [31], which are used widely to evaluate the video quality, fail to incorporate the importance of ROI. This is because these metrics weigh all pixels of the video equally, regardless of the ROI impact on user awareness [32]. For this reason, an additional video quality metric – Weighted PSNR (WPSNR) – is also used in this paper to evaluate the quality of videos with ROI [22], which is defined as [33]:

$$\begin{aligned} \text{WPSNR} &= 10log_{10}(255^2/D_{frame}) \\ D_{frame} &= \boldsymbol{\alpha} * \text{MSE}\left(\text{f}, \text{f}'\right) + (1-\boldsymbol{\alpha}) * \text{MSE}\left(\text{f}, \text{f}'\right) \end{aligned} \quad (3)$$

where MSE stands for the Mean Squared Error between the original frame and after truncation while $\boldsymbol{\alpha}$ (alpha) is defined as the weight that the ROI would have. The $\boldsymbol{\alpha}$ value will be a constant value of 0.9 following the previous research in [22]. WPSNR is a metric that can be used to evaluate quality for videos with ROI. Videos without ROI information are evaluated using both PSNR and SSIM [32].

C. SYSTEM- AND CIRCUIT-LEVEL IMPLEMENTATION

The hardware system platform from Fig. Fig. 5 implemented an H264 decoder synthesized into a Xilinx Zynq XC7Z010 FPGA fabric. The H264 decoder IP Core was designed using the Xilinx Vivado 2019.2 [34] software design suite. This same decoder is modified to include an MBTM. The FPGA was commanded via an ARM Cortex-A9 Processor running on a Linux Operating System through a custom baseband driver.

The circuit-level frame buffer is implemented using a 45nm CMOS technology [35]. The supply voltage is 1.0V. The memory size is 128 words at 24 bits per word.

D. VIDEO QUALITY EVALUATION

All selected videos were analyzed using an in-house custom software tool. The tool operated in the following three-step process: (i) Load one original video frame from memory; (ii) Apply both the method in [7] and the proposed method to the original frame and generate the truncated frame using each method; and (iii) Compare the frames generated against the original frame and calculate the PSNR, SSIM and WPSNR values. With data points collected on a per-frame basis, the average PSNR, SSIM and WPSNR of each video stream was calculated and compared.

TABLE 2. Summary of system overhead/cost using the proposed method at 1920 × 1080 resolution.

	Description	Statistics
Bitrate: Server to Mobile Device	Additional bits transmitted from server to mobile device for protocol per frame	1,200 bits or a 0.02% increase on average per frame
Network Overhead	Additional time needed for additional protocol data to transmit per frame over the 4G LTE network	Between 240µs and 100µs more time per frame on 4G LTE at 5Mbps and 12Mbps
Logic Gates: Mobile Device	Additional Look Up Tables (LUTs) needed on the FPGA implementing the Proposed Method on the mobile device	204 LUTs or a 0.38% increase in area
Power: Frame Buffer	Power savings from the frame buffer in percentage	Between 6.67% and 19.74% power decrease depending on truncation amount

E. STATISTICAL HYPOTHESIS VALIDATION

From the proposed method, the following hypothesis was conjectured: the differences between the method in [7] and the proposed method follow a Weibull distribution. This should hold true for power savings and different video quality parameters including PSNR, SSIM, and WPSNR. To support this hypothesis, a goodness of fit regression test was preformed to determine if the data falls within the probability plot of a Weibull distribution. The intention behind this analysis was to identify patterns in the output videos that serve to estimate the differences in quality and noise for any given input video. If the data follows this hypothesis, this would suggest that the sample set of videos is of adequate size and as a result, no more videos would need to be tested.

VI. EXPERIMENTAL RESULTS

A. IMPLEMENTATION OVERHEAD

Fig. 10 and Fig. 11 show the post-implementation results of the baseline H264 decoder and the H264 decoder modified to include an MBTM. When comparing both figures, one observes that the Lookup Table (LUT) overhead, which is the additional logic gates required for the proposed design over the baseline, was 204 LUTs or a 0.38% increase in area. The I/O, which was used for the server-to-mobile device interface, increased by 37, or 29.6%. The power consumption of the modified decoder also increased by 0.068 watts or 0.03%: most of which was attributed to the increased number of I/O. Finally, the Worst Negative Slack (WNS) increased by 0.011ns, which was within acceptable tolerance for this system as any positive value means that the critical path passes timing constraints. Overall, this additional overhead was negligible when compared against the benefits in power savings and quality improvements achieved using the proposed technique.

Table 2 summarizes the implementation overhead of the proposed technique with a video resolution of 1920 × 1080, which is the maximum resolution supported by the system. It can be seen that the proposed technique comes at a cost of bitrate, network overhead, and logic gate overhead. Specifically, the mobile device needs to receive 1,200 additional bits on average per frame from the server, which is a 0.02% increase on average per frame, and increases network uptime by 240μs to transmit the protocol's additional bits. Also, it needs 204 additional LUTs to implement the proposed method, which results in 0.38% area overhead. The primary advantage of the proposed method is the power savings enabled by the decoder frame buffer memory, which will be discussed in detail as follows.

B. CIRCUIT-LEVEL FRAME BUFFER TIMING DIAGRAM

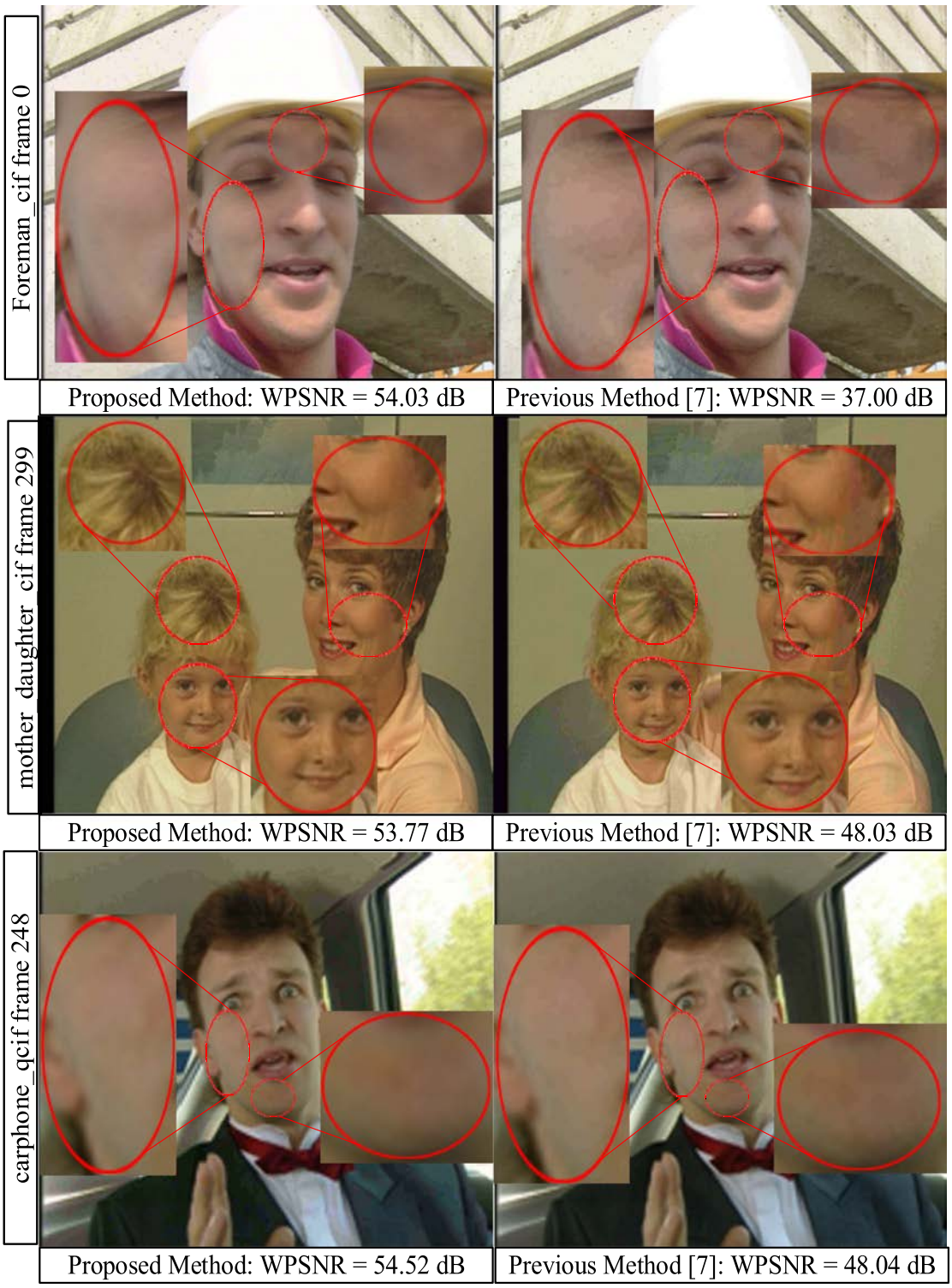
The proposed frame buffer is shown in Fig. 8 and the simulation timing diagram is shown in Fig. 9. In this waveform, phi2b, T_Y, y!, and y_pre! denote the pre-charge (for un-truncated bits), bit truncation enable for Y byte, power supply for truncated bitcell's (last 3 LSBs of Y byte), and pre-charge deactivated signal for truncated bit cells, respectively. T_U and T_V controlled the bit truncation for U and V bytes respectively. Write and read enable signals initiated the write and read operations for the memory accordingly. Data [23:0] were the three bytes of each word of the proposed memory buffer. Here, blue to red lines stand for "don't care" regions. The red lines denote where the rising clock edge was initiated for write and read operations. Finally, the green lines denote that the write and read operations were enabled. All 8 truncation permutations and traditional read and write operations were presented in the timing diagram as an exhaustive simulation of the frame buffer circuit.

It should be noted that, if the bit truncations were initiated, then 3 LSBs were truncated from the selected byte/bytes based on the control signals T_Y, T_U, and T_V. During the read operations, the 3-LSBs of the truncated bytes would output "100" bits through the utilization of 2:1 multiplexers instead of being read from memory to save power.

C. CIRCUIT-LEVEL FRAME BUFFER POWER SAVING ANALYSIS

Fig. 12 presents the power consumption of the proposed frame buffer in all eight possible conditions, including seven truncation cases and one baseline case without bit truncation. Specifically, the eight cases include: (i) No truncation with control signals T_Y & T_U & T_V = '0', (ii) Y vector truncation with T_Y = '1', (iii) U vector truncation with T_U = '1', (iv) V vector truncation with T_V = '1', (v) Y and U vectors truncation with T_Y & T_U = '1', (vi) U and V vectors truncation with T_U & T_V = '1', (vii) V and Y vectors truncation with T_V & T_Y = '1', and (viii) YUV vectors truncation with T_Y & T_U & T_V = '1'.

TABLE 3. Visual comparison of selected video frames.



As discussed in Section III-B, for the truncated vectors, the three LSB will be truncated to "100" to maximize power savings. All 8 truncation cases presented in Fig. 12 are tested in 6 ways: when written ('0' to '0', '0' to '1', '1' to '0', '1' to '1') and when read back ('0' & '1'). The power consumed in each case was calculated, and then the average is presented.

At first, a random word was initialized with $(A5A5A5)_{16}$, then the same memory word was immediately read back with $(F0F0F0)_{16}$, then all the '1's and '0's written and read received the same priority in the power consumption calculations. The same word consumed 3.90E-4 W power without any bit truncation. When the circuitry selected any

TABLE 4. Select *ROI* videos analysis results.

Videos with ROI	Truncated bits			Normalized power consumption			WPSNR (Alpha = 0.9)		
	Ref. [7]	Proposed	diff	Ref. [7]	Proposed	diff	Ref. [7]	Proposed	diff
akiyo_cif	30,412,800	32,922,081	**8.25%**	90.97%	93.55%	**-2.83%**	57.53	55.14	**-4.16%**
claire_qcif	50,079,744	44,009,266	**-12.12%**	90.97%	94.76%	**-4.16%**	57.61	57.10	**-0.88%**
dinner_1080p30	1,969,920,000	1,629,113,461	**-17.30%**	90.97%	95.07%	**-4.50%**	57.32	58.21	**1.56%**
grandma_qcif	88,197,120	78,023,685	**-11.53%**	90.97%	94.73%	**-4.12%**	57.57	57.21	**-0.62%**
intros_422_cif	36,495,360	43,295,433	**18.63%**	90.97%	92.93%	**-2.15%**	57.45	55.15	**-4.01%**
Johnny_1280x720_60	553,881,600	456,595,131	**-17.56%**	90.97%	95.09%	**-4.52%**	57.07	58.44	**2.39%**
KristenAndSara_1280x720_60	553,881,600	488,596,289	**-11.79%**	90.97%	94.74%	**-4.14%**	56.99	57.37	**0.65%**
miss_am_qcif	15,206,400	10,966,364	**-27.88%**	90.97%	95.70%	**-5.20%**	57.60	58.54	**1.63%**
news_cif	30,412,800	36,162,420	**18.91%**	90.97%	92.91%	**-2.13%**	57.52	54.48	**-5.27%**
rush_hour_1080p25	1,036,800,000	1,134,324,798	**9.41%**	90.97%	93.48%	**-2.75%**	57.49	55.78	**-2.97%**
sign_irene_cif	54,743,040	64,431,060	**17.70%**	90.97%	92.98%	**-2.21%**	57.48	55.01	**-4.29%**
trevor_qcif	15,206,400	16,565,578	**8.94%**	90.97%	93.50%	**-2.78%**	57.31	54.99	**-4.05%**
vidyo1_720p_60fps	553,881,600	597,801,678	**7.93%**	90.97%	93.57%	**-2.85%**	57.54	56.05	**-2.58%**
west_wind_easy_1080p	1,181,952,000	1,086,498,282	**-8.08%**	90.97%	94.52%	**-3.90%**	56.81	55.68	**-1.99%**
720p50_mobcal_ter	928,972,800	1,325,076,458	**42.64%**	86.60%	82.99%	**4.17%**	47.88	54.16	**13.12%**
720p50_shields_ter	928,972,800	1,245,591,004	**34.08%**	86.60%	84.01%	**2.99%**	47.69	54.48	**14.25%**
aspen_1080p	2,363,904,000	3,041,639,112	**28.67%**	86.60%	84.66%	**2.24%**	47.92	54.97	**14.71%**
blue_sky_1080p25	899,942,400	1,060,438,048	**17.83%**	86.60%	85.95%	**0.75%**	47.66	55.10	**15.61%**
bowing_cif	60,825,600	76,915,166	**26.45%**	86.60%	84.92%	**1.94%**	48.02	53.69	**11.81%**
bridge_close_cif	405,504,000	560,890,106	**38.32%**	86.60%	83.51%	**3.57%**	47.95	54.13	**12.88%**
carphone_qcif	77,451,264	88,390,034	**14.12%**	86.60%	86.39%	**0.24%**	48.04	54.52	**13.48%**
controlled_burn_1080p	2,363,904,000	2,937,098,762	**24.25%**	86.60%	85.18%	**1.63%**	47.96	55.18	**15.06%**
crew_4cif	121,651,200	172,691,140	**41.96%**	86.60%	83.07%	**4.07%**	47.54	53.54	**12.61%**
crowd_run_1080p50	2,073,600,000	3,005,452,078	**44.94%**	86.60%	82.72%	**4.48%**	47.42	53.55	**12.93%**
deadline_cif	278,581,248	334,134,316	**19.94%**	86.60%	85.70%	**1.04%**	48.02	54.48	**13.45%**
FourPeople_1280x720_60	1,107,763,200	1,318,759,148	**19.05%**	86.60%	85.80%	**0.92%**	47.96	55.12	**14.94%**
Lecture_1080P-412e	1,034,726,400	998,909,402	**-3.46%**	86.60%	88.49%	**-2.18%**	48.07	57.01	**16.91%**
life_1080p30	3,421,440,000	4,187,455,252	**22.39%**	86.60%	85.41%	**1.38%**	48.04	54.99	**14.46%**
mother_daughter_cif	60,825,600	79,283,536	**30.35%**	86.60%	84.46%	**2.47%**	48.03	53.78	**11.95%**
pamphlet_cif	60,825,600	83,561,818	**37.38%**	86.60%	83.62%	**3.44%**	47.93	53.27	**11.14%**
paris_cif	215,930,880	302,557,572	**40.12%**	86.60%	83.29%	**3.82%**	47.88	53.81	**12.40%**
pedestrian_area_1080p25	1,555,200,000	1,749,179,384	**12.47%**	86.60%	86.59%	**0.01%**	48.05	55.59	**15.70%**
rush_field_cuts_1080p	2,363,904,000	3,080,806,912	**30.33%**	86.60%	84.46%	**2.47%**	47.86	54.09	**13.01%**
salesman_qcif	91,035,648	127,208,586	**39.73%**	86.60%	83.34%	**3.77%**	47.97	53.74	**12.03%**
station2_1080p25	1,298,073,600	1,803,902,208	**38.97%**	86.60%	83.43%	**3.66%**	47.79	53.71	**12.39%**
students_cif	204,171,264	283,317,790	**38.76%**	86.60%	83.45%	**3.63%**	47.92	53.64	**11.93%**
sunflower_1080p25	2,073,600,000	2,874,084,316	**38.60%**	86.60%	83.47%	**3.61%**	47.81	53.79	**12.52%**
suzie_qcif	30,412,800	31,039,198	**2.06%**	86.60%	87.83%	**-1.42%**	48.07	55.79	**16.07%**
touchdown_pass_1080p	2,363,904,000	2,715,649,830	**14.88%**	86.60%	86.30%	**0.35%**	47.94	55.38	**15.52%**
tractor_1080p25	2,861,568,000	4,031,161,306	**40.87%**	86.60%	83.20%	**3.92%**	47.50	53.67	**12.99%**
vidyo3_720p_60fps	1,107,763,200	1,368,042,822	**23.50%**	86.60%	85.27%	**1.53%**	48.11	54.54	**13.36%**
vidyo4_720p_60fps	1,107,763,200	1,206,225,090	**8.89%**	86.60%	87.02%	**-0.48%**	48.13	55.78	**15.89%**
720p50_parkrun_ter	1,393,459,200	2,074,332,336	**48.86%**	82.11%	73.37%	**10.64%**	36.71	53.47	**45.63%**
720p5994_stockholm_ter	1,669,939,200	2,434,415,832	**45.78%**	82.11%	73.93%	**9.97%**	35.49	53.38	**50.41%**
ducks_take_off_1080p50	3,110,400,000	4,661,102,586	**49.86%**	82.11%	73.20%	**10.86%**	36.36	53.30	**46.61%**
football_422_cif	109,486,080	161,366,445	**47.39%**	82.11%	73.64%	**10.32%**	36.54	53.96	**47.67%**
football_cif	79,073,280	114,873,738	**45.28%**	82.11%	74.02%	**9.86%**	36.58	53.79	**47.03%**
foreman_cif	91,238,400	123,714,882	**35.60%**	82.11%	75.75%	**7.75%**	37.01	54.04	**46.01%**
hall_monitor_cif	91,238,400	136,539,606	**49.65%**	82.11%	73.23%	**10.82%**	36.66	53.56	**46.12%**
harbour_4cif	182,476,800	268,811,991	**47.31%**	82.11%	73.65%	**10.31%**	36.34	53.91	**48.34%**
ice_4cif	145,981,440	183,650,610	**25.80%**	82.11%	77.50%	**5.62%**	35.22	52.91	**50.21%**
mobile_calendar_422_cif	109,486,080	163,476,849	**49.31%**	82.11%	73.29%	**10.74%**	36.37	53.06	**45.88%**
old_town_cross_420_720p50	1,382,400,000	2,060,977,467	**49.09%**	82.11%	73.33%	**10.69%**	36.44	53.60	**47.07%**
riverbed_1080p25	1,555,200,000	2,285,697,270	**46.97%**	82.11%	73.71%	**10.23%**	36.62	53.29	**45.54%**
silent_cif	91,238,400	130,669,137	**43.22%**	82.11%	74.38%	**9.41%**	36.70	53.46	**45.64%**
soccer_4cif	182,476,800	249,118,389	**36.52%**	82.11%	75.58%	**7.96%**	36.49	53.36	**46.25%**
tennis_sif	45,619,200	67,173,204	**47.25%**	82.11%	73.66%	**10.29%**	36.27	54.29	**49.69%**
tt_sif	34,062,336	50,343,861	**47.80%**	82.11%	73.56%	**10.41%**	36.16	54.24	**49.98%**
vtc1nw_422_ntsc	109,486,080	146,642,766	**33.94%**	82.11%	76.04%	**7.39%**	36.67	53.74	**46.55%**
washdc_422_ntsc	109,486,080	163,223,193	**49.08%**	82.11%	73.33%	**10.69%**	36.48	53.62	**46.99%**
AVE			**26.46%**	**86.27%**	**83.79%**	**3.06%**			**20.17%**

T_Y, T_U or T_V control option, where 3-LSBs were truncated from each one selected, 6.67% power was saved when compared against no bit truncation. When T_Y & T_U, T_Y & T_V or T_U & T_V were selected, where 3-LSBs

TABLE 5. Select results of *non-ROI* videos.

Videos without ROI	Truncated bits			Normalized power consumption			PSNR (dB)			SSIM		
	Ref. [7]	Proposed	diff	Ref. [7]	Proposed	diff	Ref. [7]	Proposed	diff	Ref. [7]	Proposed	diff
bus_cif	30,412,800	43,042,560	**41.53%**	86.60%	82.47%	4.77%	48.15	40.82	**7 dB**	0.9966	0.9779	**-0.0188**
galleon_422_cif	72,990,720	102,643,456	**40.63%**	86.60%	83.23%	3.89%	48.2	40.72	**7 dB**	0.9956	0.9707	**-0.025**
highway_cif	405,504,000	569,610,752	**40.47%**	86.60%	83.25%	3.87%	48.24	41.14	**7 dB**	0.9925	0.9536	**-0.0392**
tempete_cif	52,715,520	77,495,808	**47.01%**	86.60%	83.12%	4.01%	48.12	40.81	**7 dB**	0.997	0.9725	**-0.0246**
bridge_far_cif	638,972,928	958,070,016	**49.94%**	82.11%	73.18%	10.88%	42.56	40.6	**2 dB**	0.966	0.9453	**-0.0214**
city_4cif	182,476,800	271,175,040	**48.61%**	82.11%	73.42%	10.59%	42.48	40.84	**2 dB**	0.9869	0.9702	**-0.0169**
coastguard_cif	91,238,400	120,874,752	**32.48%**	82.11%	76.30%	7.08%	42.48	41.07	**1 dB**	0.9877	0.9735	**-0.0144**
container_cif	91,238,400	125,445,888	**37.49%**	82.11%	75.41%	8.17%	42.45	41.01	**1 dB**	0.9766	0.9669	**-0.0099**
flower_cif	76,032,000	107,439,360	**41.31%**	82.11%	74.72%	9.00%	42.5	40.93	**2 dB**	0.9929	0.9844	**-0.0086**
flower_garden_422_cif	109,486,080	162,798,336	**48.69%**	82.11%	73.40%	10.61%	42.57	40.62	**2 dB**	0.9894	0.9782	**-0.0113**
garden_sif	34,974,720	52,448,256	**49.96%**	82.11%	73.18%	10.88%	42.52	40.73	**2 dB**	0.9911	0.9813	**-0.0098**
husky_cif	76,032,000	111,882,240	**47.15%**	82.11%	73.68%	10.27%	42.48	40.91	**2 dB**	0.9949	0.9784	**-0.0166**
mobile_cif	91,238,400	136,164,864	**49.24%**	82.11%	73.31%	10.73%	42.57	40.7	**2 dB**	0.9925	0.9855	**-0.0071**
waterfall_cif	79,073,280	118,609,920	**50.00%**	82.11%	73.17%	10.89%	42.4	40.63	**2 dB**	0.9883	0.9775	**-0.011**
AVE			**44.61%**	**83.40%**	**76.56%**	**8.26%**	**44.12**	**40.82**	**3dB**	**0.9891**	**0.9726**	**-0.0167**

TABLE 6. Comparision with prior work.

	6T/8T SRAM [8]	Heterogeneous sizing SRAM [10]	Split-data SRAM [9]	Data-dependent SRAM [37]	SRAM with hamming [11]	SRAM with ECC [40]	Viewer-aware memories [4, 5, 6]	Content-aware memory [7]	**This Work**
Quality runtime adaptation	No	No	Yes	Yes	Yes	Yes	Yes	Yes	**Yes**
Considering viewer's experience	No	No	No	No	No	No	Yes	Yes	**Yes**
Video content adaptation	No	No	No	No	No	No	No	Yes	**Yes**
ROI awareness	No	No	No	No	No	No	No	No	**Yes**
Bitcell	6T and 8T	Larger 6T	8T and 10T	10T	6T	6T	6T	6T	**6T**
Induced bitcell area overhead	11% overhead[1]	30% overhead[2]	18.5%-52% overhead[3]	over 30% overhead[4]	0% overhead	0% overhead	0% overhead	0% overhead	**0% overhead**

[1]With 3 8T bitcells in a byte
[2]With optimized bitcell sizes for a byte
[3]52% overhead with 3 10T in a byte without video data bit truncation and 18.5% overhead with video data bit truncation
[4]The exact area overhead of 10T over 6T was not provided in the paper. Each 10T consists of a traditional 8T and two extra transistors and the area overhead of 8T is approximately 30%, so the overhead of 10T is larger than 30%.

were truncated from each selected byte, then 13.33% power was saved when compared against no bit truncation. Finally, when T_Y, T_U, and T_V were selected for truncation, where 3-LSBs were truncated from each selected byte, then 19.74% power was saved. The supply voltage for this simulation was 1V, where the proposed frame buffer circuit can operate to specification and had no faulty bit(s).

D. VIDEO VISUAL QUALITY COMPARISONS

Table 3 shows visual frame comparisons for three selected videos with ROI between the proposed method and [7]. It can be seen that the proposed technique enables significant visual quality improvement as compared to [7]. Specifically, for the Foreman_cif video, due to the truncated LSBs in [7], the man's cheeks, forehead, and hat shadows experience noticeable banding distortion, negatively affecting video quality. Alternatively, the proposed ROI-aware technique effectively reduces the banding distortion and improves the visual quality. Similarly, with [7], the mother_daughter_cif demonstrates banding distortion around the cheeks and hair, and the carphone_qcif video suffers from discoloration around the cheeks and chin. The introduced ROI awareness of the proposed technique effectively avoids losing the quality of videos. Another observation from Table 3 is that the proposed technique achieves a much higher WPSNR value of all three videos. A more detailed analysis on WPSNR will be provided in next sub-section.

E. OBJECTIVE VIDEO QUALITY AND BIT TRUNCATION ANALYSIS

Table 4 compares WPSNR values and the number of truncated bits of 60 videos with ROI using the proposed technique to the state-of-the art [7]. As shown, the proposed technique can enable 26.46% additional truncated bits as compared to [7]. Meanwhile, with the ROI awareness, the proposed technique can effectively enhance the quality of

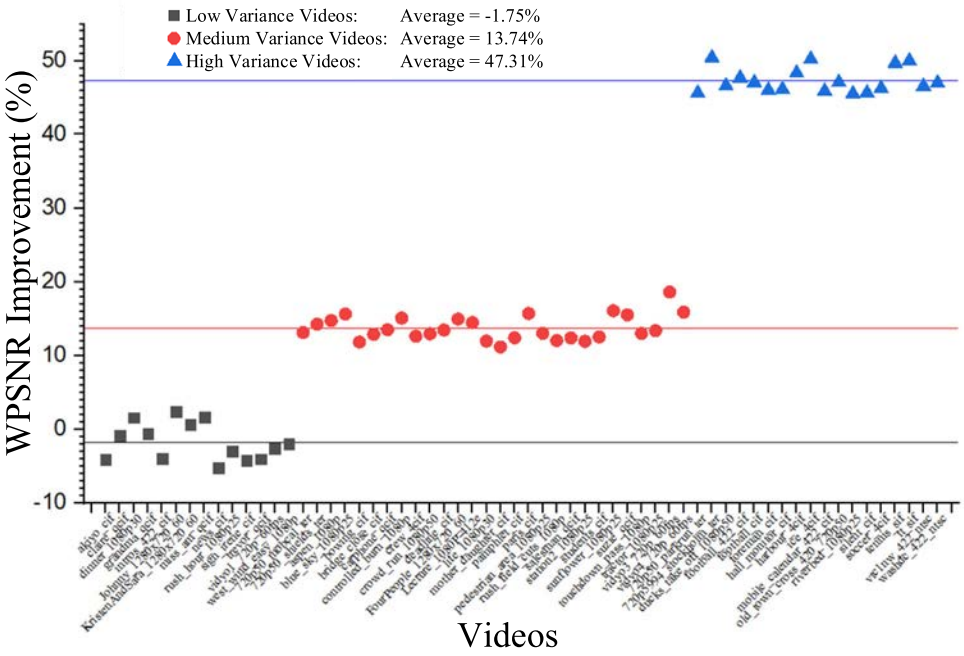


FIGURE 13. Impact of Select video content characteristics on the effectiveness of the proposed technique, compared to old technique.

the majority of videos. On average, the proposed technique can increase the WPSNR values by 20.17%, as compared to [7].

We further analyzed the impact of the MB variance characteristics (low, medium, and high variance) on the effectiveness of the proposed technique. The results are shown in Fig. 13. As can be seen, the WPSNR improvement strongly depends on the MB variance of videos. Specifically, videos with high variance achieve the most significant quality improvement using the proposed technique, with 47.31% WPSNR increase on average. With the proposed technique, all videos with medium variance also demonstrate quality improvement, with 13.74% WPSNR increase on average. However, the proposed technique shows little video quality improvement for videos with low variance and even results in minimal video quality degradation (with 1.75% WPSNR loss on average. This suggests that the proposed technique is particularly effective for videos with high and medium MB variance.

We compared the video quality of 14 videos without ROI using the proposed technique and the state-of-the art [7]. As shown in Table 5, the proposed technique can enable a much larger number of truncated bits, with quality degradation as compared to [7]. On average, 44.61% additional truncated bits can be achieved for videos without ROI, with 3dB PSNR loss. To further assess the video quality loss induced by the proposed technique, the SSIM value of each video was evaluated. As observed in Table 5, the average SSIM with proposed technique and the design [7] is 0.9726 and 0.9891, respectively. Using the proposed technique, for the majority videos (13 out of 14), the SSIM value is much higher than 0.95, which is the acceptable SSIM threshold value [36]; only one video (i.e., bridge_far_cif) results in a SSIM of 0.9453, which is very close to the acceptable SSIM value (i.e., 0.95 [36]). It can be concluded that the proposed technique can result in significant number of truncated bits while delivering an acceptable perceived quality.

F. VIDEO-LEVEL POWER SAVING ANALYSIS

To compare the power effectiveness of the proposed ROI-aware technique to the traditional memory design and the state-of-the art [7], we model the power consumption of the memory for a video as:

$$P\left(Video_i\right) = \frac{1}{N_i}\sum_{j=1}^{N_i} P_k\left(j\right) \quad k \in (0, 1, 2, 3) \tag{4}$$

where N_i is the total number of bytes for the video i, $P_k(j)$ is the normalized power consumption to store byte j with k truncated bits. For the proposed memory, $k = 3$; for the traditional memory, $k = 0$; for the memory in [7], $k = 0, 1, 2,$ or 3. For a fair comparison, the normalized power consumption $P_k(j)$ is based on the power consumption reported in [9]. The results are listed in Table 6 and Table 5. As observed, the proposed technique only consumes 83.79% and 76.56% total power on average for videos with ROI and videos without ROI, respectively, as compared to the traditional memory. Also, the proposed technique achieves 3.06% and 8.26% power savings for videos with ROI and videos without ROI, respectively, as compared to [7]. It is worth mentioning that, our analysis only considers the facial features as ROI of videos and integrating advanced ROI

identification algorithms will covert videos without ROI to videos with ROI, thereby further increasing the effectiveness of our proposed technique to general videos.

G. STATISTICAL ANALYSIS

In order to verify that the selected video analysis results are a representation of the full population of all videos, we further carried out a statistical analysis based on 889 different videos, as discussed in Section V-A. Specifically, the Pearson's Chi-square test [37], which is also known as the Chi-Squared goodness-of-fit test, is used in our analysis. The *goodness-of-fit* test checks whether the sample data is likely to be from a specific theoretical distribution, and therefore represents the data expected in the actual population. The idea is, if the sample data does fit an expected distribution, then it shows that the sample data represents the full population of the video data in existence. The statistical results will either reject or accept the working statement called the *null hypothesis*, H0, which is the opposite of the *alternative hypothesis*, H1. To reject or accept the null hypothesis, several methods exist, one of which is the Probability value method i.e. *P-Value method*. The *P-Value* is the evidence against the null hypothesis, i.e., the smaller the P-Value, the stronger the evidence that the null hypothesis should be rejected. The P-Value method is based on a critical value, which is determined based on the distribution. For example, if we are dealing with a *normally distributed* population – which we are according to our statistical results shown later, this critical value is a *z-score*. The z-score is a value that is then used to lookup the P-Value in a Standard Normal *z-table*, which is used to then test the null hypothesis. If a P-Value is greater than an alpha or α value of 0.05, then the statistical results are ''not significant'' and thus, the null hypothesis is accepted. However, if the PValue is less than or equal to α values of 0.05 or 0.01, then the results are ''significant'' or ''highly significant'' respectively, and thus, the null hypothesis is rejected in favor of the alternative hypothesis. The rejection regions depend on the confidence level that the results are significant, e.g., if a confidence level is 95%, then an α value of 5% or 0.05 is chosen: 100% - 95%.

In our analysis, the null hypothesis for the Chi-Squared goodness-of-fit test, H_0, is, ''For the given set of video data points, a specified distribution accurately represents the data'', and therefore, the alternative hypothesis, H_1, is, ''For the given set of video data points, a specified distribution *does not* accurately represent the data.'' Hence, the goal of the statistical analysis is to validate the null hypothesis and thus deduce that the specified distribution would fit the data. To achieve this statistical result, P-values were calculated for three different video categories – low, medium, and high variance – for power savings and three different video quality metrics (i.e., PSNR, SSIM, and WPSNR). In our analysis, we used the MathWave Technologies EasyFit software [38], to identify the Chi-Squared goodness-of-fit test, in order to determine the type of distribution.

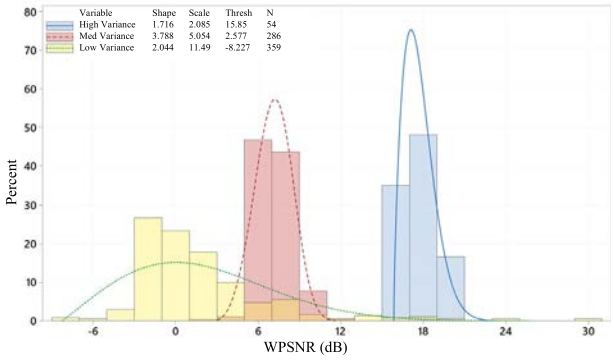


FIGURE 14. Histogram of SSIM distributions between the truncation method in [7] and the proposed method shown. With Weibull distribution parameters and Number of data points. All distributions that fall within a 99% Confidence Interval.

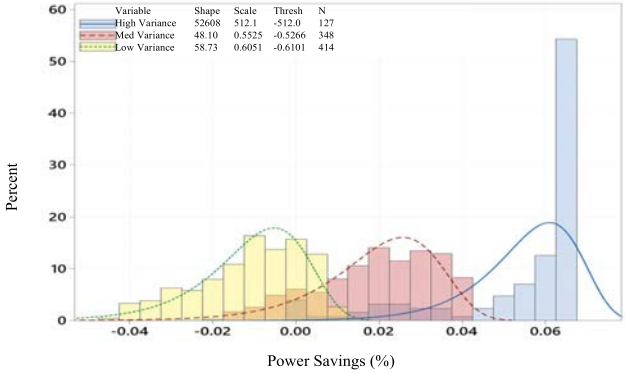


FIGURE 15. Histogram of power savings, measured in percentage improvement, between the truncation method in [7] and the proposed method. With Weibull distribution parameters and Number of data points. All distributions that fall within a 99% Confidence Interval.

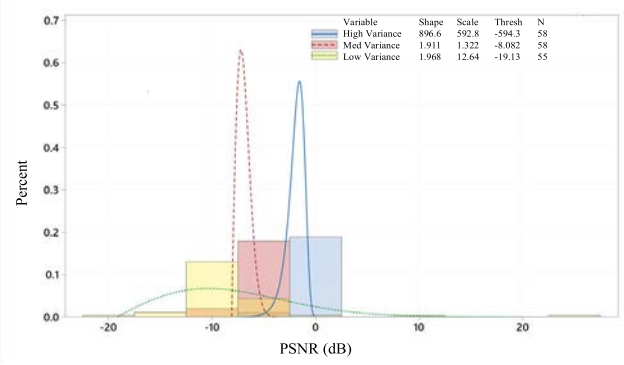


FIGURE 16. Histogram of quality distributions between the truncation method in [7] and the proposed method shown. With Weibull distribution parameters and Number of data points. All distributions that fall within a 99% Confidence Interval.

The results are shown in Fig. 14-16. Specifically, Fig. 14 demonstrates how categorizing the data creates clear groupings when comparing the truncation method in [7] to the proposed method. The figure shows three distinct 3-parameter Weibull distributions that describe the quality improvement between the proposed and [7]. These Weibull distributions are within the 99% confidence interval required. All distribution reports a P-value greater than 0.98,

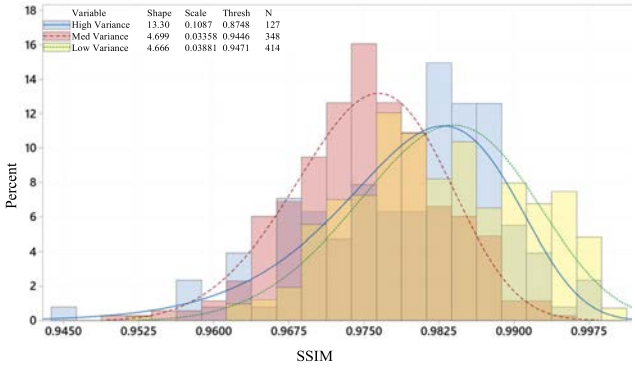


FIGURE 17. Histogram of PSNR noise increase between the truncation method in [7] and the proposed method shown. With Weibull distribution parameters and Number of data points. All distributions that fall within a 99% Confidence Interval.

implying that we cannot reject the null hypothesis and accept this distribution as a possible representation of the data. Fig. 15 shows the power savings distribution for each video type as a 3-parameter Weibull distribution. Power savings is reported as a percentage increase, using the total number of bits truncated in each video and the power consumption shown in Fig. 12. All of these distributions pass the 99% confidence interval. Fig. 17 shows the probability of noise increase in a random video stream. All distributions shown fall into the category of Weibull distributions with a 99% confidence interval. Fig. 14 shows the probability of quality drop measured in SSIM for a random video stream, with a 99% confidence interval that the probability lies within a Weibull distribution. The most notable differences between Fig. 17 and Fig. 14 is that the Medium and Low Variance videos show very little difference for the SSIM loss, whereas the loss is very distinct in the PSNR distribution The preformance of the proposed method was better according to SSIM as compared to PSNR, since PSNR preforms the comparison between the entire image without considering the user's perception.

It was determined that because all videos are compared to themselves for improvement, e.g., video after the proposed method is applied verses the original video, video resolution has no statistical impact in the data set. Power Consumption will be presented by improvement percentage, thus ignoring linear growth in watts saved in larger scale videos. Similarly, it is statistically sound that a larger dataset is not needed to affirm the distributions. As all distributions shown fall within the 99% confidence interval, there is only a 1% chance that the data collected is far from the specified distribution.

In summary, videos categorized as high variance show the biggest improvements in WPSNR quality, the most power saving by percentage, and introduce the least noise as measured by PSNR and SSIM. With medium variance videos also saving on power consumption, with a more noticeable drop in quality and increase in noise. As such, videos classified as low variance often have little to gain using this method, and sometimes even cause video quality degradation.

VII. COMPARISON WITH PRIOR WORK

Table 6 compares this work against state-of-the art low-power video memory designs. As shown, our proposed memory enables more-flexible run-time power-quality adaptation according to video content characteristics of each frame, while considering the important region within one frame from a perceptual point of view.

A. COMPARED TO STATE-OF-THE-ART APPROXIMATE VIDEO MEMORIES

To enhance the power efficiency of video storage, approximate video-specific memories have been developed to store the MSBs of video data in more robust memory bitcells, such as more-than-6T SRAM bitcells [8], [9], upsized 6T [10]. In order to minimize the video quality loss, those techniques typically store MSBs in reliable bitcells (e.g., 8T, 10T, or upsized 6T), thereby leading to a tolerable output quality degradation. However, those approximate video memory designs usually bring large implementation overhead (e.g., up to 52% [9]). More importantly, for those techniques, the achieved video quality is fixed during design-time, so they lack of adaptation at run-time to meet different requirements of a variety of video applications.

B. COMPARED TO STATE-OF-THE-ART ADAPTIVE VIDEO SRAM

Recently, in order to enable run-time power-quality adaptation, several video SRAM designs have been presented, such as data-dependent memory [39], SRAM with selective hamming (15,11) [11], and SRAM with error-correction-code (ECC) adaptation [40]. The data-dependent SRAM consists of 10T bitcells and associated conditional pre-charge circuitry to adapt to the stored data's statistical dependencies. The developed SRAM with selective hamming (15,11) [11] can switch between no ECC and hamming (15,11) based on the quality requirement of the video applications. The very recent SRAM design with ECC adaptation [40] supports three different power-quality tradeoff levels, including hamming code-74, hamming code-1511, and no ECC. However, those adaptive memory designs focus on efficiency optimization while maintaining an acceptable video quality, such as keeping the PSNR values above 30 dB [11], [40], and they did not consider the viewer's experience in the memory design process. Therefore, they may cause large and inefficient design margins.

C. COMPARED TO STATE-OF-THE-ART VIEWER-AWARE VIDEO MEMORY

By introducing viewer's experience to video memory design process, we have studied that memory failures can be leveraged to improve video system power efficiency without sacrificing viewer's experience [4]–[6]. The basic idea is that in high noise-tolerance viewing contexts with

high-illuminance levels, memory failures are intentionally introduced by adaptively disabling LSBs of the video data stored in memories. This line of studies illustrates a new dimension of power savings for hardware design through the introduction of memory failures. However, those designs did not consider the variance of different videos and they are not sufficient to support videos with various content characteristics.

D. COMPARED TO STATE-OF-THE-ART CONTENT-AWARE VIDEO MEMORY

The content-aware SRAM presented in [7] is another recent viewer-aware memory design that can enable run-time power-quality adaptation based on the content characteristics of video applications. However, it adapts the number of truncated LSBs of video data based on the average plain macroblock percentage of an entire video sample, so it is not suitable for the videos with frame-level difference. Section VI provided detailed comparison results and analysis between the proposed memory and memory presented in [7], in terms of power savings, video quality, number of truncated bits. It concludes that the proposed memory enables an average of 26.46% additional truncated bits, 3.06% power savings, and 20.17% WPSNR improvement for ROI videos, as compared to in [7].

E. COMPARISON SUMMARY

In our developed video memory technique, the ROI regions are identified and utilized to enable intelligent tradeoff between video quality and power efficiency of video storage in mobile devices. Accordingly, the proposed memory enables run-time quality adaptation with significantly reduced pixel bits and further power savings, as compared to existing techniques. To the best of our knowledge, this is the first work that can adapt the video storage to frame-level video content and important regions from viewer's perceptual experience point of view. Our proposed ROI-aware video memory is orthogonal to existing viewer-aware or data-dependent schemes and therefore can be simultaneously utilized to further optimize power efficiency.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we have presented a video content-adaptable Region-of-Interest (ROI)-aware video storage technique to optimize the power efficiency. The ROI of videos is identified and protected to preserve the video quality, while other regions are truncated with 3-LSB truncation for power savings. To support the proposed method, a low-power frame buffer was developed that implemented 3-LSB truncation which enabled runtime quality and power adaptation. Our results show that the proposed technique only uses 83.79% and 76.56% of the power on average for videos with ROI and without ROI respectively, as compared to the traditional memory and the state-of-the art [9], respectively. Meanwhile, the proposed technique can increase the quality (i.e. WPSNR values) by 20.17% on average for the videos with ROI and 26.46% additional truncated bits as compared to [9]. For the videos without ROI, the proposed technique can realize 44.61% additional truncated bits and 8.26% power savings as compared to [9], while maintaining a healthy above 40dB PSNR and 0.95 SSIM. This paper focuses on the facial features as ROI of videos; our future investigations would include extensions of ROI identification to deal with general videos. Additionally, psychological experiments will be conducted to access the visual experience of viewers for hardware optimization.

REFERENCES

[1] *Mobile Data Traffic Outlook*. Accessed: Feb. 28, 2022. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-forecast

[2] T. Liu, S. Wang, W. Lee, J. Yang, K. Hou, and C. Lee, "A 125μW, fully scalable MPEG-2 and H.264/AVC video decoder for mobile applications," *IEEE J. Solid-State Circuits*, vol. 42, no. 1, pp. 161–169, Jan. 2007.

[3] F. Sampaio, M. Shafique, B. Zatt, S. Bampi, and J. Henkel, "Energy-efficient architecture for advanced video memory," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD)*, Nov. 2014, pp. 132–139.

[4] D. Chen, X. Wang, J. Wang, and N. Gong, "VCAS: Viewing context aware power-efficient mobile video embedded memory," in *Proc. 28th IEEE Int. Syst.–Chip Conf. (SOCC)*, Beijing, China, Sep. 2015, pp. 333–338.

[5] D. Chen, J. Edstrom, L. Yang, M. E. McCourt, J. Wang, and N. Gong, "Viewer-aware intelligent efficient mobile video embedded memory," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 4, pp. 684–696, Apr. 2018.

[6] J. Edstrom, D. Chen, J. Wang, H. Gu, E. A. Vazquez, M. E. McCourt, and N. Gong, "Luminance-adaptive smart video storage system," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 734–737.

[7] J. Edstrom, Y. Gong, A. A. Haidous, B. Humphrey, M. E. Mccourt, Y. Xu, J. Wang, and N. Gong, "Content-adaptive memory for viewer-aware energy-quality scalable mobile video systems," *IEEE Access*, vol. 7, pp. 47479–47493, 2019.

[8] I. J. Chang, D. Mohapatra, and K. Roy, "A priority-based 6T/8T hybrid SRAM architecture for aggressive voltage scaling in video applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 2, pp. 101–112, Feb. 2011.

[9] N. Gong, S. Jiang, A. Challapalli, S. Fernandes, and R. Sridhar, "Ultra-low voltage split-data-aware embedded SRAM for mobile video applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 12, pp. 883–887, Dec. 2012.

[10] J. Kwon, I. Lee, and J. Park, "Heterogeneous SRAM cell sizing for low power H.264 applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 99, no. 2, pp. 1–10, Feb. 2012.

[11] F. Frustaci, M. Khayatzadeh, D. Blaauw, D. Sylvester, and M. Alioto, "SRAM for error-tolerant applications with dynamic energy-quality management in 28 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 50, no. 5, pp. 1310–1323, Mar. 2015.

[12] A. Kazimirsky, A. Teman, N. Edri, and A. Fish, "A 0.65-V, 500-MHz integrated dynamic and static RAM for error tolerant applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 9, pp. 2411–2418, Sep. 2017.

[13] M. C. Chi, C. H. Yeh, and M. J. Chen, "Robust region-of-interest determination based on user attention model through visual rhythm analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 1025–1038, Jul. 2009.

[14] (2017). *YouTube-8M Dataset*. [Online]. Available: https://research.google.com/youtube8m/

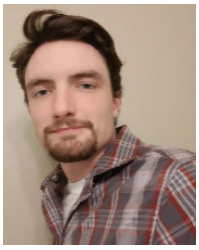
[15] I. Himawan, W. Song, and D. Tjondronegoro, "Automatic region-of-interest detection and prioritisation for visually optimised coding of low bit rate videos," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2013, pp. 76–82.

[16] Y. Huo, X. Wang, P. Zhang, J. Jiang, and L. Hanzo, "Unequal error protection aided region of interest aware wireless panoramic video," *IEEE Access*, vol. 7, pp. 80262–80276, 2019.

[17] Y.-F. Ma, X.-S. Hua, L. Lu, and H.-J. Zhang, "A generic framework of user attention model and its application in video summarization," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 907–919, Oct. 2005.

[18] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to OpenCV," in *Proc. 35th Int. Conv. MIPRO*, Opatija, Croatia, May 2012, pp. 1725–1730.
[19] X.-W. Tang, X.-L. Huang, F. Hu, and Q. Shi, "Human-perception-oriented pseudo analog video transmissions with deep learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9896–9909, Sep. 2020.
[20] M. Shafique, "Application-guided power-efficient fault tolerance for H.264 context adaptive variable length coding," *IEEE Trans. Comput.*, vol. 66, no. 4, pp. 560–574, Apr. 2017.
[21] M. Shafique, B. Molkenthin, and J. Henkel, "An HVS-based adaptive computational complexity reduction scheme for H.264/AVC video encoder using prognostic early mode exclusion," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2010, pp. 1713–1718.
[22] Y. Liang, H. Wang, and K. El-Maleh, "Design and implementation of content-adaptive background skipping for wireless video," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 4–7.
[23] R. P. Foundation. *Raspberry Pi Documentation*. [Online]. Available: https://www.raspberrypi.org/documentation/
[24] Xilinx. *Z-Turn Board (With Zynq-7020)*. Accessed: Nov. 1, 2020. [Online]. Available: https://www.xilinx.com/products/boards-and-kits/1-571ww1.html
[25] MAGEWELL. *USB Capture Utility V3*. Accessed: Nov. 1, 2020. [Online]. Available: http://www.magewell.com/usb-capture-utility-v3
[26] Ylonen, T. Rinne, and Tatu. *SCP(1)–Linux Man Page*. Accessed: Apr. 14, 2013. [Online]. Available: https://linux.die.net/man/1/scp
[27] Q. Bin. *Osen Logic OSD10 H.264 Decoder*. Accessed: 2018. [Online]. Available: http://bbs.eetop.cn/viewthread.php?tid=628991
[28] I. E. Richardson, *The H.264 Advanced Video Compression Standard*, 2nd ed. West Sussex, U.K.: Wiley, 2010.
[29] *Xiph.Org Video Test Media [Derf's collection]*. Xiph. Accessed: Oct. 18, 2020. [Online]. Available: https://media.xiph.org/video/derf/
[30] Y. Wang, S. Inguva, and B. Adsumilli, "YouTube UGC dataset for video compression research," in *Proc. IEEE 21st Int. Workshop Multimedia Signal Process. (MMSP)*, Kuala Lumpur, Malaysia, Sep. 2019, pp. 1–5.
[31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[32] S. Ma, J. Q. Qu, and L. and Zhang, "Overall visual quality optimization coding for region of interest," in *Proc. APSIPA ASC*, 2011, pp. 1–6.
[33] J. Erfurt, C. R. Helmrich, S. Bosse, H. Schwarz, D. Marpe, and T. Wiegand, "A study of the perceptually weighted peak signal-to-noise ratio (WPSNR) for image compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, Taiwan, Sep. 2019, pp. 2339–2343.
[34] Xilinx. (2019). *Vivado Design Suite*. [Online]. Available: https://www.xilinx.com/products/design-tools/vivado.html
[35] *FreePDK45*. Accessed: Mar. 4, 2022. [Online]. Available: https://eda.ncsu.edu/freepdk/freepdk45/
[36] M. Zanforlin, D. Munaretto, A. Zanella, and M. Zorzi, "SSIM-based video admission control and resource allocation algorithms," in *Proc. 12th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2014.
[37] K. Pearson, "Karl Pearson, paper on the chi square goodness of fit test (1900)," in *Landmark Writings Western Math. 1640–1940*. Amsterdam, The Netherlands: Elsevier, 2005, ch. 56, pp. 724–731.
[38] *EasyFit*. Accessed: Mar. 7, 2022. [Online]. Available: https://easyfit.informer.com/download/
[39] C. Duan, A. J. Gotterba, M. E. Sinangil, and A. P. Chandrakasan, "Energy-efficient reconfigurable SRAM: Reducing read power through data statistics," *IEEE J. Solid-State Circuits*, vol. 52, no. 10, pp. 2703–2711, Oct. 2017.
[40] H. Das, A. A. Haidous, S. C. Smith, and N. Gong, "Flexible low-cost power-efficient video memory with ECC-adaptation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 10, pp. 1693–1706, Oct. 2021.

ALI HAIDOUS (Member, IEEE) received the B.S. degree in electrical and computer engineering (ECE) from Lipscomb University, in 2015, and the M.Eng. and Ph.D. degrees in ECE from North Dakota State University (NDSU), in 2017 and 2021, respectively. He is currently a Licensed Professional Engineer. His research interests include embedded hardware–software system integration, video memory optimization for low-power applications, and content aware power efficient smart memory systems.



WILLIAM OSWALD received the B.S. degree in computer engineering and the M.S. degree in electrical engineering from the University of South Alabama, in 2020 and 2021, respectively. He is currently pursuing the Ph.D. degree in systems engineering. He has worked in industry as a Systems Analyst at Packaging Corporation of America, from 2019 to 2020. His research interests include computer architecture design, machine learning, and neural networks.

HRITOM DAS (Member, IEEE) received the B.S. degree in electrical and electronic engineering from American International University-Bangladesh, Dhaka, Bangladesh, in 2012, the M.Sc. degree in electronic engineering from Kyungpook National University, Daegu, South Korea, in 2015, and the Ph.D. degree in electrical and computer engineering from North Dakota State University, Fargo, ND, USA, in 2020. He is currently a Visiting Assistant Professor with the Department of Electrical and Computer Engineering, University of South Alabama, Mobile, AL, USA. His research interests include the low power circuit design, testing, AI implementation on traditional electronics, and smart computer vision.

NA GONG (Member, IEEE) received the B.E. degree in electrical engineering and the M.E. degree in microelectronics from Hebei University, Hebei, China, in 2004 and 2007, respectively, and the Ph.D. degree in computer science and engineering from The State University of New York, Buffalo, in 2013. She is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of South Alabama, Mobile, AL, USA. Her research interests include power-efficient computing circuits and systems, memory optimization, and neuromorphic hardware.