

LOW POWER ENTROPY CODING HARDWARE DESIGN FOR H.264/AVC BASELINE PROFILE ENCODER

Chuan-Yung Tsai, Tung-Chien Chen, and Liang-Gee Chen

DSP/IC Design Lab, Graduate Institute of Electronics Engineering,
National Taiwan University, Taipei, Taiwan; Email: cytsai@video.ee.ntu.edu.tw

ABSTRACT

Low power hardware design for entropy coding of H.264/AVC baseline profile encoder is urgent for the increasing mobile applications. However, previous works are poor in the power performance. In this paper, the first low power Context-based Adaptive Variable Length Coding (CAVLC) scheme named the Side Information Aided (SIA) Symbol Look Ahead (SLA) one-pass CAVLC is proposed, with the non-zero and abs-one SIA flags. A reconfigurable architecture for the SLA module is also proposed to support the low power CAVLC scheme efficiently. The resultant hardware power is reduced by 69% to only 3.7mW at 27MHz and 1.8V for CIF-sized video coding. The total logic gate count is 27K gates.

1. INTRODUCTION

H.264/AVC [1] is the new generation video coding standard developed by the Joint Video Team. It can save about 25%–45% bit-rate compared to MPEG-4 Advanced Simple Profile (ASP). The ultra high coding efficiency comes from many new features, including the inter prediction with variable block sizes and multiple reference frames, the intra prediction, and the Context-based Adaptive Variable Length Coding or Binary Arithmetic Coding (CAVLC or CABAC). Because of the outstanding performance, its applications in mobile devices are increasing rapidly. When designing an H.264/AVC baseline profile encoder for low-delay mobile applications, the low power considerations are strongly necessary for its hardware accelerators, including the Entropy Coding (EC) engine.

Since CAVLC is the key tool of EC in H.264/AVC baseline profile encoder, many hardware designs [2–5] have been proposed. However, due to the large power consumption in the residual SRAM and CAVLC symbol buffers, previous EC hardware designs have poor power performances. It is also difficult to reduce the EC power based on existing CAVLC schemes and architectures, because of the new context-based adaptivity and the residual data transmission scheme.

In this paper, we present a low power EC hardware design for the H.264/AVC baseline profile encoder. Based on the proposed low power CAVLC scheme and architecture, the EC hardware power consumption are reduced significantly compared to previous works. The rest of this paper is organized as

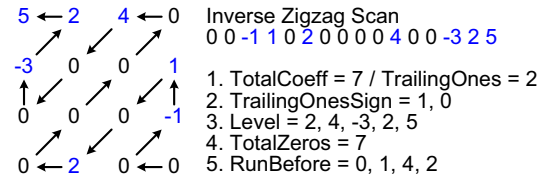


Fig. 1. Example of a 4×4 residual block's CAVLC procedure.

follows. In Section 2, we will introduce the EC fundamental knowledge and define the problems. The low power CAVLC scheme and EC architecture are proposed in Section 3 and 4 respectively. The implementation results are shown in Section 5. Finally, Section 6 gives the conclusion.

2. FUNDAMENTALS AND PROBLEM STATEMENT

In H.264/AVC baseline profile encoder, EC consists of two encoding tools—Exp-Golomb coding and CAVLC, which are for the Macro-Block (MB) headers and transformed prediction residuals respectively. Because Exp-Golomb coding has far lower power consumption compared to CAVLC, only the introduction of CAVLC will be given in this section. For more details of Exp-Golomb coding, please refer to [1].

The CAVLC defines six encoding symbols, including the *TotalCoeff*, *TrailingOnes*, *TrailingOnesSign*, *Level*, *TotalZeros*, and *RunBefore*. An example of a 4×4 residual block (the transformation & coding unit) CAVLC procedure is shown in Fig. 1. First, all residuals are scanned in the inverse zigzag order for extracting the symbols. The definitions for all symbols are listed below. It is worth to note that the *TotalCoeff* and *TrailingOnes* form a joint symbol for coding.

Symbol Definition

- *TotalCoeff*: number of total non-zero residuals.
- *TrailingOnes*: number of consecutive ± 1 from the beginning, considering only the non-zero residuals.
- *TrailingOnesSign*: sign of *TrailingOnes*, one for negative and zero for positive.
- *Level*: value of non-zero residuals.
- *TotalZeros*: zeros after the first non-zero residual.

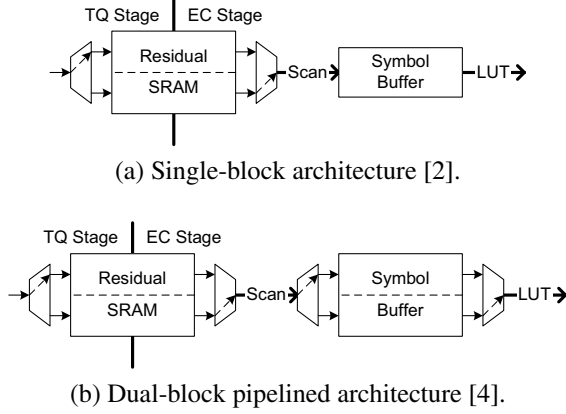


Fig. 2. Block diagrams of previous scan-and-LUT two-pass CAVLC scheme.

- *RunBefore*: zeros between current and previous non-zero residuals, recorded from the second non-zero residual till all zeros have been counted.

The *TotalCoeff* symbol, as an important factor of context-based adaptivity in CAVLC, is the first one among all symbols to be coded. This is because other CAVLC symbols are coded with adaptive table selection depending on the value of *TotalCoeff*, which can help to compress the statistical redundancies efficiently. In the previous two-pass CAVLC schemes, the CAVLC engine first needs to sequentially scan all transformed prediction residuals in the residual SRAM, and on-line extracts the symbols into buffers. This is called the scan pass. Worth to note, the buffers cannot be discarded because all symbols must wait the *TotalCoeff* symbol for the context-based adaptive coding. Only after the scan pass is finished, can the *TotalCoeff* be obtained and the Look-Up-Table (LUT) pass start to code symbols. This forms the single-block two-pass CAVLC [2] as shown in Fig. 2(a). In order to increase the throughput, a dual-block pipelined architecture as shown in Fig. 2(b) was proposed in [4], which can perform the scan and LUT simultaneously.

However, these two previous schemes are poor in the low power performance due to their large power consumption in the residual SRAM and register-based symbol buffers. Since their residual data transmission scheme is based on the MB pipelining [6] system architecture, residuals passed from the Transformation-Quantization (TQ) stage are in general solely stored in SRAM. Thus a complete SRAM scan is always necessary even though the probability of non-zero residuals is usually low for general video sequences (only 2.5% in Foreman sequence with QP=30.) Similarly, utilization of the symbol buffers containing a large number of registers is also very low. It is even worse for the dual-block pipelined scheme which has double registers. Due to these characteristics, the power performance can hardly be improved without more fundamental optimizations in the CAVLC scheme.

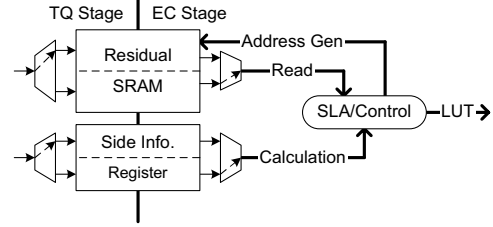


Fig. 3. Block diagram of SIA SLA one-pass CAVLC scheme.

Residuals	Non-Zero Flags	Abs-One Flags
5 2 4 0	1 1 1 0	0 0 0 0
-3 0 0 1	1 0 0 1	0 0 0 1
0 0 0 -1	0 0 0 1	0 0 0 1
0 2 0 0	0 1 0 0	0 0 0 0

Fig. 4. Example of proposed SIA flags definition.

3. PROPOSED SCHEME

In this paper, we propose the Side Information Aided (SIA) Symbol Look Ahead (SLA) one-pass CAVLC scheme which can help to reduce the large power consumption in the residual SRAM and symbol buffers very efficiently. Its block diagram is shown in Fig. 3. The side information registers can help to minimize the residual SRAM accesses by indicating the locations of non-zero residuals, and meanwhile provide more information for one-pass symbol look ahead and coding. This means the symbols can be obtained and immediately coded in one pass, before a complete scan of the residual SRAM.

The first type of adopted side information is called *non-zero flags*, which are composed of a MB-sized 2D array of one-bit registers. The value of its entry is assigned to 1 if the corresponding residual is non-zero valued, and 0 is assigned otherwise. The second side information type is *abs-one flags*. If the residual value equals ± 1 , its corresponding entry is set to 1. An example of assigning the SIA flags is shown in Fig. 4. With the non-zero and abs-one flags, the low power CAVLC scheme is proposed, and listed as follows. It is worth to note that the SIA flags are also accessed in the inverse zigzag scan order (as defined in Fig. 1) during executing the scheme.

Proposed Scheme

- *TotalCoeff*: sum up all non-zero flags.
- *TrailingOnes*: sum up the abs-one flags located before the first one of XOR of non-zero and abs-one flags.
- *TrailingOnesSign*: read residual SRAM with address generated by the non-zero flag's position.
- *Level*: same as *TrailingOnesSign*.
- *TotalZeros*: sum up the inverted non-zero flags located after the first non-zero flag.
- *RunBefore*: subtract previous non-zero flag's position from current non-zero flag's position.

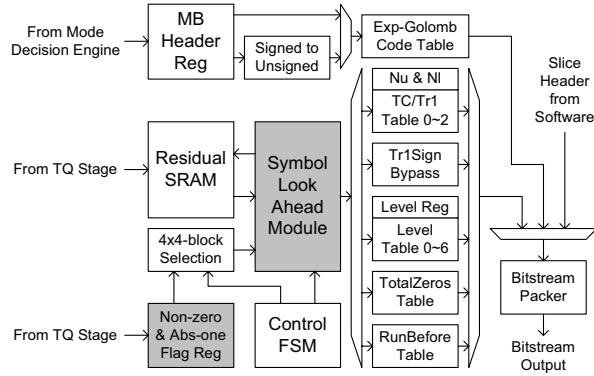


Fig. 5. Proposed low power architecture of H.264/AVC baseline profile EC. Shaded modules are the key features.

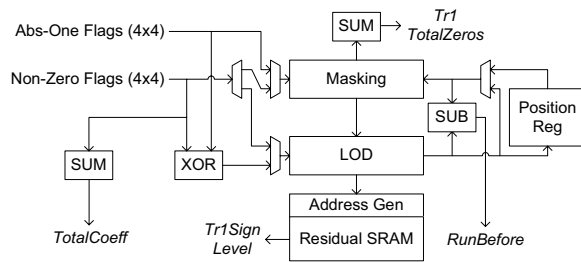


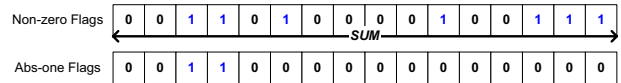
Fig. 6. Proposed reconfigurable SLA module.

The non-zero flags is greatly helpful to improving the total CAVLC performance. Most symbols can be obtained from it, or from the residual SRAM with reading address which is easily generated by its information. The abs-one flags in co-operation with the non-zero flags can make the *TrailingOnes* symbol coding also one-pass. Without abs-one flags, a residual SRAM scan is required to calculate the number of trailing ones among all residuals, and the SLA scheme will fail. With both two flags as the side information, the low power CAVLC scheme can minimize the residual SRAM accesses and eliminate the symbol buffers, such that the total EC power can be greatly reduced.

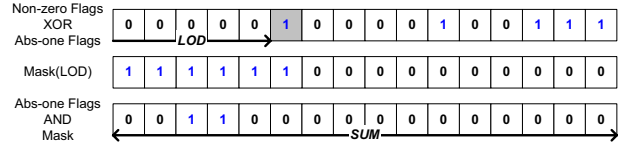
4. ARCHITECTURE

Figure 5 shows the proposed low power EC architecture. In this work, the Exp-Golomb coding engine for MB header & CAVLC engine for prediction residuals are both implemented as hardware. The encoding of much less frequent slice header will be handled by software. The Exp-Golomb coding for MB header of H.264/AVC involves only some simple conversions and LUT operations. After Exp-Golomb coding, the CAVLC engine will then start to encode the residual blocks. In the CAVLC engine, an SLA module is implemented with the SIA flags as input to execute the symbol look ahead operations. All symbols are extracted and fed to the LUT modules in one

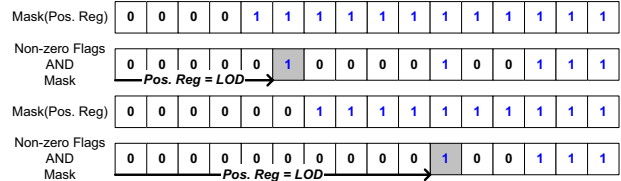
1. Non-zero & abs-one flags and generation of *TotalCoeff*



2. Generation of *TrailingOnes*



3. Address generation for *Tr1Sign* or *Level* (two consecutive cycles)



4. Generation of *TotalZeros*

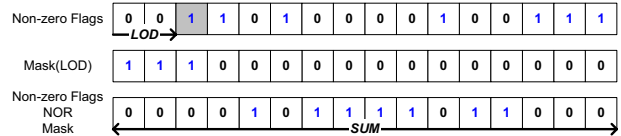
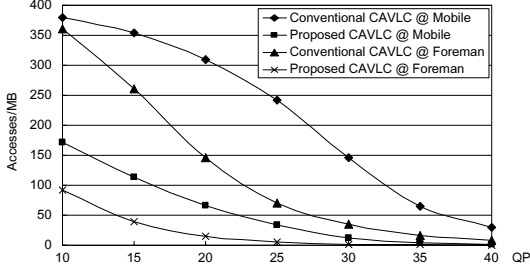


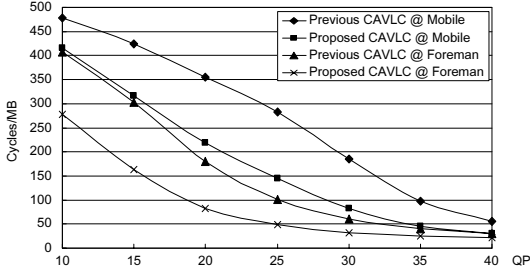
Fig. 7. Example of SLA module data flows. The binary string rows are ordered according to each configuration's data flow.

pass without buffering. Finally, the bitstream packer packs all coded bits into a bus-width-aligned bitstream for output.

The proposed SLA module in this paper is the key component for supporting the low power CAVLC scheme. Figure 6 shows the proposed reconfigurable architecture. It is designed to exploit the hardware resource sharing of common circuits in the SLA module. A Leading One Detector (LOD) is employed to generate the value of first one's position in a 16-bit binary string. The Masking Unit (MU) generates a 16-bit binary mask and perform bitwise logic operations between the mask and its input. An example of the SLA module's data flows is shown in Fig. 7. The SLA module has two configurations. The first one is for the *TrailingOnesSign*, *Level*, and *RunBefore*, with data path from MU through LOD to the Position Register (PR). The second one is for the *TrailingOnes* and *TotalZeros* with data path from LOD to MU. We use Fig. 7(3) to explain the the first configuration's data flow as an example. It starts with PR containing 4, which means the fourth residual has been detected as a non-zero residual. In the first row of Fig. 7(3), MU generates a mask of 4 leading zeros and 12 trailing ones according to PR. Then in the second row, the bit-wise AND operation masks all detected non-zero flags into zeros, such that the LOD can detect the next non-zero flag's position, map it to non-zero residual's address, and update PR. The third and fourth rows show the work repeated in the next cycle. Please note that the unshown *RunBefore*



(a) Memory accesses per MB compared with [2][4].



(b) Processing cycles per MB compared with [4].

Fig. 8. Simulation results of performance comparisons.

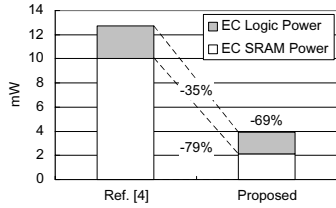


Fig. 9. Comparison of power performances between [4] and proposed design for Mobile-Calendar sequence with QP=30.

flow is the same as Fig. 7(3), except the output is switched to current LOD result minus PR value.

5. IMPLEMENTATION RESULTS

In order to evaluate the performance of proposed low power CAVLC scheme, the residual SRAM access frequencies are simulated and compared in Fig. 8(a). About 85% of SRAM accesses are saved according to the results of four different CIF sequences, which means the residual SRAM power can be greatly reduced. The throughput of proposed EC hardware also exceeds the previous high-throughput design [4], and the results (in processing cycles per MB) are shown in Fig. 8(b).

The proposed EC hardware is implemented in Verilog and synthesized using the Synopsys Design Compiler with TSMC Artisan 0.18 μ m cell library. The total power consumption is only about 3.7mW at 27MHz and 1.8V, which is simulated with Synopsys PrimePower at CIF sequences with QP set to 30 (medium bit-rate). Compared to [4], about 69% of total power reduction is achieved as shown in Fig. 9. The logic gate

Table 1. Gate Count Profile of Proposed EC Hardware

Component	Gate Count
Look-Up-Table	3758
SLA module	6160
Bitstream Packer	3317
Controller (Core/AHB)	4806
Non-zero & Abs-one Flags	8557
Total	26598

count of proposed low power EC hardware design is 26,598 gates, and its profile is listed in Table 1.

6. CONCLUSION

In this paper, we propose a low power EC hardware design for H.264/AVC baseline profile encoder. With the SIA SLA one-pass CAVLC scheme in cooperation with non-zero & abs-one SIA flags, the total EC power consumption is greatly reduced. Besides, an efficient reconfigurable architecture for the SLA module is proposed. The resultant total EC power is reduced by 69% to only 3.7mW at 27MHz and 1.8V for CIF sequences at medium bit-rate. The total logic gate count is 27K gates.

7. REFERENCES

- [1] Joint Video Team, *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification*, ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC, May 2003.
- [2] Y. W. Huang, B. Y. Hsieh, T. C. Chen, and L. G. Chen, "Hardware architecture design for H.264/AVC intra frame coder," in *Proc. of Int. Symposium on Circuits and Systems (ISCAS)*, 2004, pp. 269–72 Vol.2.
- [3] I. Amer, W. Badawy, and G. Jullien, "Towards MPEG-4 part 10 system on chip: a VLSI prototype for context-based adaptive variable length coding (CAVLC)," in *Proc. of IEEE Workshop on Signal Processing Systems (SIPS)*, 2004, pp. 275–9.
- [4] T. C. Chen, Y. W. Huang, C. Y. Tsai, B. Y. Hsieh, and L. G. Chen, "Dual-block-pipelined VLSI architecture of entropy coding for H.264/AVC baseline profile," in *Proc. of IEEE VLSI-TSA Int. Symposium on VLSI Design, Automation and Test (VLSI-TSA-DAT)*, 2005, pp. 271–4.
- [5] Y. K. Lai, C. C. Chou, and Y. C. Chung, "A simple and cost effective video encoder with memory-reducing CAVLC," in *Proc. of Int. Symposium on Circuits and Systems (ISCAS)*, 2005, pp. 432–5 Vol.1.
- [6] T. C. Chen, Y. W. Huang, and L. G. Chen, "Analysis and design of macroblock pipelining for H.264/AVC VLSI architecture," in *Proc. of Int. Symposium on Circuits and Systems (ISCAS)*, 2004, pp. 273–6 Vol.2.