VLSI ARCHITECTURE AND IMPLEMENTATION OF A HIGH-SPEED ENTROPY DECODER

Ming-Ting Sun

Bellcore, 331 Newman Springs Rd., Red Bank, NJ 07701, U.S.A.

Abstract

In many important applications such as image, video, and facsimile coding, entropy coding is used to exploit the statistics of the input data to achieve lossless data compression. The most often used entropy coding techniques include variable-length coding and run-length coding. Many proposed video standards have also included the variable-length coding and run-length coding as standard techniques in their entropy coding algorithms. In this paper, an improved Variable-Length Decoder (VLD) architecture which can achieve higher throughput in decoding variable-length codes compared to previously reported VLD architecture is presented. Besides the new VLD architecture, an experimental research prototype VLSI (Very Large Scale Integration) implementation of an entropy decoder which includes the VLD and Run-Length Decoder (RLD) is also discussed. The chip will be fabricated using an 1 μm double-metal CMOS technology. It is to be used in an experimental research prototype HDTV (High Definition Television) codec with a sample rate of 52 MHz. The chip contains about 46,000 transistors in a die size of about 5 mm x 5 mm.

1. Introduction

In many important applications such as image, video, and facsimile coding, entropy coding is used in the encoder to exploit the statistics of the input data to achieve lossless data compression. The most often used entropy coding techniques include variable-length coding [1] and run-length coding [2]. In variable-length coding, shorter codewords are assigned to more probable source symbols so that the average bit-rate is reduced. In run-length coding, consecutive zeros are represented by the zero-run-length so that the number of samples is reduced. An entropy decoder is required in the decoder to decode the variable-length and run-length codes to recover the original data. Many proposed video standards [3-5] have also included the variable-length coding and run-length coding as standard techniques in their entropy coding algorithms.

In a typical HDTV codec for broadband ISDN applications, the sampling frequency is higher than 50 MHz and the final compressed data rate is about 130 Mb/s [6]. At these rates, as will be discussed in the next section, it is very difficult to implement an entropy decoder consisting of a Variable Length Decoder (VLD) and Run Length Decoder (RLD). In this paper, an improved VLD architecture which can achieve higher throughput in decoding variable-length codes compared to previously reported architectures is presented. Besides the new VLD architecture, an experimental research prototype VLSI (Vary Large Scale Integration) implementation of an entropy decoder which includes the VLD and RLD is also discussed. The chip will be fabricated using an 1 μm double-metal CMOS technology. It is designed to be used in an HDTV codec with a sample rate of 52 MHz. Based on simulation results, the chip can also operate at a much higher rate. At the 52 MHz rate, the VLD handles a worst case input rate of 832 Mb/s and a constant output rate of 416 Mb/s. The chip contains about 46,000 transistors in a die size of about 5 mm x 5 mm.

The organization of the paper is as follows. Section 2 discusses the difficulties of designing a high-speed entropy decoder. In Section 3, the improved VLD architecture is described. Section 4 presents the VLSI implementation of the entropy decoder. The conclusions are summarized in Section 5.

2. The Entropy Decoder

A block diagram of the commonly used entropy coding system which uses variable-length coding and run-length coding is shown in Fig.1. At the encoder side, the data go through the Run-Length Coder and Variable-Length Coder. Since the data rate at the output of the Variable-Length Coder is highly irregular, a rate-smoothing buffer memory is used to smooth out the data rate. The buffer memory has a fixed width (eg. 16-bit). The Variable-Length Coder concatenates the coded data bits into a serial bit-stream. When the length of the bit-stream exceeds the fixed width of the buffer memory, the Variable-Length Coder outputs a word with the fixed width to the buffer memory. At the decoder side, the operations are reversed. It should be noted that since the Variable-Length Coder uses longer codelength for less frequent symbols, at certain time, the Variable-Length Coder may result in data expansion rather than data compression. As an example, in an HDTV codec with 8-bit input data at a 52 MHz constant rate and a maximum codelength of 16-bit, the Variable-Length Coder and Decoder have to be able to handle a variable data-rate up to 832 Mb/s.

In the entropy encoder, there is no feedback loop involved. Thus, it is relatively easy to apply extensive pipelining to achieve high-speed operations. However, it is difficult to design the entropy decoder for real-time high-speed applications such as HDTV. The difficulties are due to the following reasons: (1) After variable-length codes are concatenated to form the serial data stream, there are no longer explicit word boundaries. Thus, we do not know the beginning of a codeword until the previous codeword is decoded. Because of this inherent feedback loop,

CH 3006-4/91/0000 – 0200 $1.00 © IEEE

parallel processing and pipelining can not be easily applied to the VLD to achieve a high throughput. (2) When the RLD is decoding a run-length code, the operation of the VLD should be stopped until the correct number of zeros has been generated by the RLD. Thus, there is another feedback loop involving the VLD and RLD. (3) The input data are supplied from the external fixed-width buffer memory. Since a word read from the external buffer may contain variable number of samples, a feedback signal is required to control the input bit stream from the buffer. This feedback signal also complicates the interface circuitry. (4) Many system issues such as detecting transmission errors and regaining synchronizations after transmission errors also need to be considered.

There are several methods to decode a stream of variable-length codewords. The straightforward method is to shift the data stream bit-by-bit, searching along the code-tree for a codeword. Since this approach processes the data stream bit-by-bit, it often needs to operate at a very high clock rate which may not be feasible for high sample-rate applications. Another approach has been recently proposed in [7-11] which processes the data word-by-word. Every codeword is decoded in one clock cycle regardless of the codeword length. Thus, high throughput can be achieved with relatively low clock-rate operations. However, in the architecture discussed in [9-11], the speed of the operation is limited by a feedback loop which includes a barrel shifter, a PLA (Programmable Logic Array), and an accumulator. In the next section, an improved VLD architecture is described. The feedback loop of the new VLD only contains a barrel shifter and a PLA. The accumulator has been moved out of the feedback loop. Thus, much higher speed variable-length decoding can be achieved.

3. A High Speed VLD Architecture

A block diagram of the proposed VLD is shown in Fig.2. The maximum codelength is assumed to be 16-bit. The circuit can be generalized to other maximum codelengths. The circuit contains a decoding part and an interface part. The decoding part can decode a variable-length code in one clock cycle regardless of the codelength. The interface part provides continuous data stream to the decoding part. It also interfaces with the external fixed-width buffer memory and the internal variable-length decoding part. The circuit operation is briefly described as follows. An example is given in Fig.3 to illustrate the decoding operation.

The decoding part contains three latches L_0, L_1, and L_2, a barrel shifter BS_0, and a PLA. The barrel shifter can shift the data stream multiple bits (from 1 to 16 bits) in one clock cycle. In the beginning of the operation, the data stream is loaded into the latch L_0. The latch L_2 is initialized to control BS_0 so that the first 16-bit of data appears at the output of BS_0 (input of the PLA) for decoding. The PLA performs a fast pattern matching. When a codeword is matched, the decoded codeword and the codelength appear at the outputs of the PLA. In the next clock cycle, the output of BS_0 is latched into L_1 so that the first bit of the previous codeword will appear at the first bit of BS_0 input. Also, the previous decoded codelength is latched into L_2 to control BS_0 to shift to the beginning of the current codeword to be decoded, and the operation repeats.

Since the data in L_1 are shifted every clock cycle, the content of L_0 also needs to be replenished every clock cycle so that the data stream in L_0 and L_1 is kept continuous. This is achieved by the circuits in the interface part. The interface part contains three latches L_3, L_4, and L_5, a barrel shifter BS_1, and an accumulator. The accumulator accumulates the decoded codelength. The output of the accumulator controls the barrel shifter BS_1 to shift the input data stream to provide the correct data for L_0. When the accumulator generates a "carry", it indicates that at least 16 bits of data have been decoded and the data in L_4 are no longer need to be kept. In this case, the content of L_3 is loaded into L_4 and a "read" signal is generated to read a new 16-bit data from external buffer memory into L_3 to provide continuous data for later decoding. The operations performed in the interface part are not in the critical loop. Thus, the timing requirement is much relaxed.

The feature of the architecture is that the operations in the feedback loop are minimized. The critical loop only contains the barrel shifter BS_0 and the PLA. In practical implementations, the PLA uses half clock cycle for precharging, and another half clock cycle for evaluation. The delay time of BS_0 and the PLA precharging are shorter than the PLA evaluation. Thus, the speed of the VLD is only limited by the speed of the PLA.

4. VLSI Implementation of the Entropy Decoder

The entropy decoder chip consists of a VLD as described in the previous section, a RLD, and an error-checking circuit. The chip also contains some test circuitry for testing purposes. In the test mode, the chip can bypass the RLD so that the intermediate results after the VLD are available at the output of the chip. Also, in the test mode, some other internal signals can be observed at the chip output.

The VLD contains six codebooks which can be used for signals with different statistics. The codebooks are mask programmable. The largest codebook in the current implementation contains 160 entries. The maximum codelength is 16-bit. The layouts of the codebooks are generated by a PLA generator written in C-Language.

The PLAs are implemented by Domino CMOS type of circuits [12-13] which use one clock phase for precharging and another clock phase for evaluation. The transistors in the circuits are properly sized to minimize the charge sharing effect in the circuit. To minimize the power consumption, the unselected PLAs are disabled.

In the actual implementation of the VLD, different ways can be used to implement the accumulator which may lead to minor modifications to the circuit shown in Fig.2. A barrel shifter based accumulator is used in the current implementation. Using a 4-bit accumulator may result in a smaller size and slightly higher speed because of the reduced capacitive loading. The reason of using the barrel shifter based accumulator is because we use the PLA generator developed in the previous work [10] in order to save design time. In the architecture used in [10], the speed of the codelength decoding is more important than the speed of the codeword decoding. The generated PLAs work with a barrel shifter based accumulator in order to optimize the codelength decoding speed.

The RLD contains a 7-bit down-counter, several registers, and some random logics. The maximum allowable zero-run-length is 128. When the most significant bit of the VLD output is "0", it indicates the decoded codeword is a magnitude code. In this case, the codeword is passed to the output registers unchanged. Otherwise, the VLD decoded codeword is a zero-run-length code. The codeword is latched into the down-counter and the RLD outputs a signal to suspend the operation of the VLD until the required number of zeros represented by the codeword has been reached.

The Error-checking circuit counts the number of decoded samples. The expected total sample count is input to the entropy decoder chip through an 8-bit input port. When the total count of the actually decoded sample is different from the expected sample count, the error-checking circuit outputs an error signal to indicate that transmission errors have been detected.

All the registers implemented in the chip are static registers. The static registers are slower than dynamic registers but allow very low frequency operations and provide higher reliability. In the case when higher speed operation is required and low-frequency operation is not a concern, dynamic registers can be used to further increase the speed of the chip.

The entropy decoder chip is laid-out using a full-custom design tool Mulga [14]. It will be fabricated using an 1-um double-metal CMOS technology. The chip is designed to be used in a system with 52 MHz sample rate. Using a circuit simulator EMU [15], the simulation waveforms of the barrel shifter BS_0 and a codebook are shown in Figure 4. The delay time of the barrel shifter (including the output buffer) is about 2.8 ns. The delay time of the PLA evaluation (also including the output buffer) is about 5.2 ns. The total delay in the critical path (including the register output delay, the register set-up time, and other delays in control gates) is about 13.2 ns which translates into an operation speed of about 75 MHz. A chip layout is shown in Fig.5. The chip contains about 46,000 transistors in a die size of about 5 mm x 5 mm. A C-program has been written to simulate and verify the architecture. The program is also used in generating test vectors to verify the layout and used in system simulations.

5. Conclusions

Designing an entropy decoder which consists of a VLD and RLD meeting the high-speed requirement of many important applications such as HDTV is a difficult problem. In the paper, a new architecture of VLD is presented. The VLD can achieve higher throughput compared to previously reported architectures. The VLSI implementation of an experimental research prototype entropy decoder chip is also discussed. The chip is designed to be used in a system with a 52 MHz sample rate, but based on simulation results, it can also operate at a much higher rate. At 52 MHz, the VLD handles an worst case input rate of 832 Mb/s and a constant output rate of 416 Mb/s. The entropy decoder chip contains about 46,000 transistors in a die size of about 5 mm x 5 mm. It provides a solution to one of the major difficulties in implementing an HDTV codec.

Acknowledgement

We would like to thank S. Palaniraj for the help in doing the layout and writing the PLA generator. We also like to thank S.M. Lei, P.E. Fleischer, M.L. Liou, and R.R. Cordell for helpful discussions during the work.

References

[1] D.A. Huffman, "A Method for the Construction of Minimum Redundancy Codes," Proc. of IRE, vol.40, pp.1098-1101, Sept. 1952.

[2] W.K. Pratt, Digital Image Processing, John Wiley & Sons, pp.632, 1978.

[3] CCITT H.261-1990, Video Codec for Audiovisual Services at px64 kbit/s, 1990.

[4] ISO/MPEG, MPEG Video Simulation Model Three (SM3), Document 90/041, July 25, 1990.

[5] ISO/JPEG, JPEG Technical Specification Revision 8, JPEG-8-R8, Aug. 14, 1990.

[6] P.E. Fleischer, T.C. Chen, and S.M. Lei, "Coding of Advanced TV for BISDN Using Multiple Subbands," Proc. of Int. Symp. on Circuits and Systems, New Orleans, Louisiana, pp.1314-1318, May 1990.

[7] J.W. Peake, "Decompaction," IBM Technical Disclosure Bulletin, vol.26, No.9, pp.4794-4797, Feb. 1984.

[8] M.E. Lukacs, "Variable Word Length Coding for a High Data Rate DPCM Video Coder," Proc. of Picture Coding Symposium, pp.54-56, 1986.

[9] M.T. Sun, K.M. Yang, and K.H. Tzou, "A High-Speed Programmable VLSI for Decoding Variable-Length Codes," Applications of Digital Image Processing XII, A.G. Tescher, editor, Proc. SPIE 1153, Aug. 1989.

[10] S.M. Lei, M.T. Sun, K. Ramachandran, and S. Palaniraj, "VLSI Implementation of an Entropy Coder and Decoder for Advanced TV Applications," Proc. of Int. Symp. on Circuits and Systems, New Orleans, Louisiana, pp.3030-3033, May 1990.

[11] S.M. Lei and M.T. Sun, "An Entropy Coding System for Digital HDTV Applications," IEEE Trans. on Circuits and Systems for Video Technology, March 1991.

[12] R.H. Krambeck, C.M. Lee, and H.S. Law, "High Speed Compact Circuits with CMOS," IEEE J. Solid-State Circuits, vol. SC-17, pp.614-619, June 1982.

[13] J.A. Pretorius, A.S. Shubat, and A.T. Salama, "Charge Redistribution and Noise Margins in Domino CMOS Logic," IEEE Trans. on Circuits and Systems, vol.CAS-33, No.8, pp.786-793, Aug. 1986.

[14] N. Weste, "MULGA - An iterative symbolic layout system for the design of integrated circuits," Bell Syst. Tech. J., vol.60, no.6, pp.823-857, July-Aug. 1981.

[15] B. Ackland and N. Weste, "Functional Verification in an Iterative Symbolic IC Design Environment," in Proc. 2nd Caltech Conf. VLSI, pp.285-298, Jan. 1981.

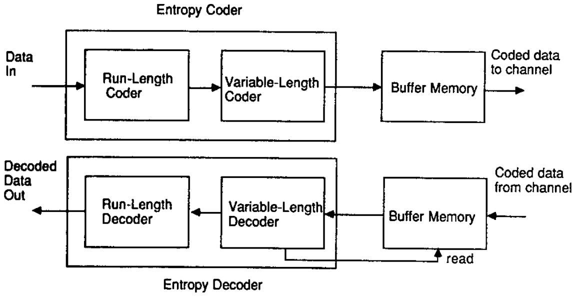

Fig.1 A block diagram of the entropy coding

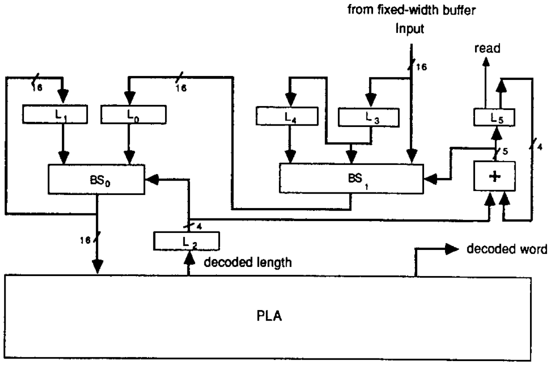

Fig.2 A Variable-Length Decoder

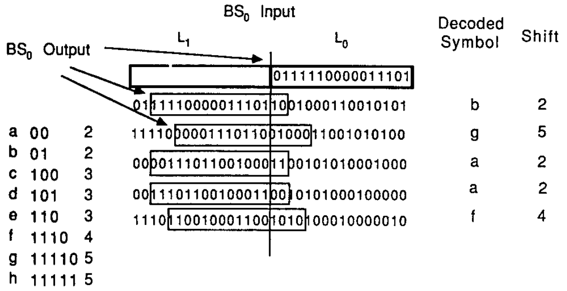
BS_0 Input
BS_0 Output
L_1
L_0
Decoded Symbol
Shift
01111110000011101
011111100000111011001000110010101 b 2
11110000011101100100011001010100 g 5
00001110110010001100101010001000 a 2
00111011001000110010101000100000 a 2
11101100100011001010100010000010 f 4
a 00 2
b 01 2
c 100 3
d 101 3
e 110 3
f 1110 4
g 11110 5
h 11111 5

Fig.3 An example illustrating the operation of the VLD

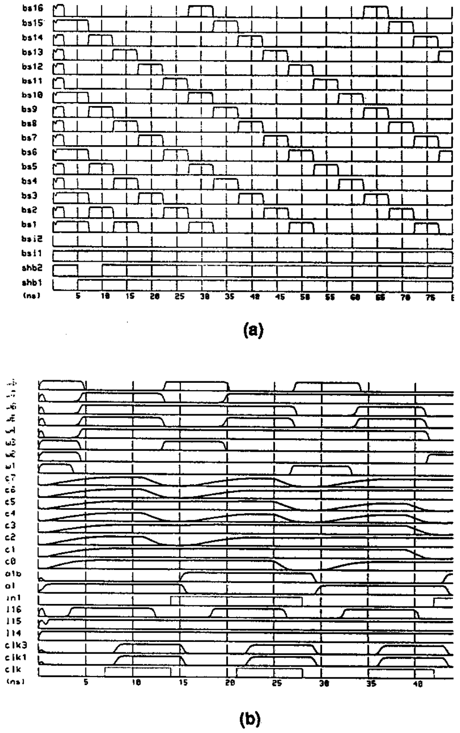

Fig.4 Simulation Waveforms:
(a) Barrel Shifter, (b) PLA

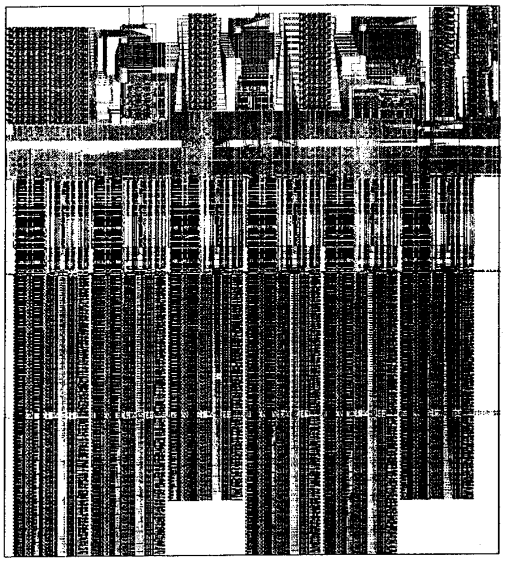

Fig.5 A mask layout of the entropy decoder chip