A Size-optimization Design for Variable Length Coding Using Distributed Logic*

SHIH-CHANG HSIA[†] and CHIEN-CHENG TSENG

*Department of Computer and Communication Engineering,
National Kaohsiung First University of Science and Technology, Kaohsiung 824, Taiwan, ROC*

(Received 1 July 1999; In final form 14 October 1999)

In this paper, we first employ an efficient approach to reduce the time to construct a codeword. With this codeword, a novel VLSI architecture is proposed to realize high-speed Variable Length Coding (VLC). In order to combine with other circuits using cell-based design, we adopt distributed logic rather than memory devices (ROM, PLA) for the implementation. In this architecture, the VLC coding scheme is partitioned into two parts, one is the codeword length and order index for bit control, another is the codeword bank for actual codeword generation. The advantage is that the circuit size of the proposed method can be reduced, where the transistor count of proposed method is only 1/2 and 1/4 of that of ROM- based and PLA-based in average, respectively.

Keywords: Codeword; VLC; PLA; ROM; Architecture; Compression; Memory

I. INTRODUCTION

Variable length coding (VLC) is a popular technique for reducing redundancy in data due to its lossless data compression. In current coding systems, VLC is also a key component that combines with other data compression method to reduce bit rate further, such as video coding systems. Efficient data encoding schemes can be valuable in the design of databases [1], and many algorithms [2, 3] are employed to realize VLC by software programs. However, they can't meet the speed requirement in

practical applications. Thus the VLSI implementation for variable length coding is announced in reports [4–10] that integrate circuits into single chips with low cost fabrication to attain high-speed operation. Most of papers only discuss the real-time VLC decoder, because the VLC encoder is not considered as the critical bottleneck in a VLC codec. Generally, traditional memory devices such ROM [4] or PAL [6] that combines with control circuits can easily realize VLC encoders for real time applications. However, when chip design employs memory devices, which circuit size can

---

* This work was supported by the National Science Council, Republic of China, under Grant NSC87-2213-E-235-001.

[†] Corresponding author. Fax: 886-7-6011012, e-mail: hsia@cems.nkfu.edu.tw

not be efficiently reduced in the implementation, and its design scheme becomes more difficult in system simulation.

It is very interesting “how to use a new architecture rather than memory device and reduce the implementation cost”. Basically, the circuit implementation with cell-based scheme is easier to create and control than memory device in the system design. With this reason, we explore the realization of a VLC by using distributed cell logic. Although logic synthesis tools are popular in current chip design, the circuit size becomes larger if we directly write “If-then” structure for VLC codeword with high level language (such as VHDL) because there are too many “if-then” statements. To achieve this goal, we first employ a new approach that is an easy and efficient method to produce codeword, the results are the same as binary tree-based. Then, a new architecture that uses distributed logic rather

than complete ROMs, PLAs structure is proposed. There are two advantages in our proposed method, one is that the circuits can be efficiently reduced, the other is that the circuits can easily combine with other system with cell-based design to take full system simulation. In order to describe this paper, the structure is organized as following: In Section II, we address a fast approach for the construction of codeword. The new architecture for variable length coding is illustrated in Section III. Section VI shows the performance comparison with existed VLC technologies.

II. THE CODEWORD CONSTRUCTION

Variable length code (VLC) is referred using Huffman coding [5], which maps the probabilities of symbols into variable-length codeword. The

FIGURE 1 Setup procedure for codeword construction.

FIGURE 2 Codeword assignment with binary tree based.

codewords are created by binary-tree structure, the setup procedure is shown in Figure 1. We first compute the probability of each symbol, and sort their probabilities. The binary-tree construction can be established by Huffman-Tree, then the codeword assignment is according to probabilities, which is shown in Figure 2.

In order to create codeword quickly, a new approach is used, which the operation is shown in Figure 3. The symbol C_k^r denotes the codeword that is in r th order of the k th group, where k is codeword length. C_k^1 , and C_k^2 represent the 1st and 2nd codewords in the k th group, respectively. In the same k th group, the codeword is sequentially increased by one, thus $C_k^{r+1} = C_k^r + 1$. When the length of codeword is increased, the first codeword can be attained from the addition of last codeword of current length and one, then the bit data is shifted by left-2. So, we have

$$C_{k+1}^1 = 2(C_k^{\max(r)} + 1) = 2T_k, \quad (1)$$

where T_k is the threshold of k length codeword. T_k can be determined by

$$T_k = \sum_{n=2}^k 2^{k-n} Y[n], \quad (2)$$

$Y[n]$ denotes the number of codeword in k th group. For example, when $k=4$, $T_4 = 2^2 Y[2] + 2 Y[3] +$

TABLE I The creation of codewords for each symbol

Symbol	Codeword	Length	r th word
a	00	$k=2$	1
b	010	$k=3$	1
c	011	$k=3$	2
d	1000	$k=4$	1
e	1001	$k=4$	2
f	10100	$k=5$	1
g	10101	$k=5$	2
h	10110	$k=5$	3
i	10111	$k=5$	4

$2^0 Y[4] = 10 = (1010)_2$. With this method, all VLC codewords could be produced, Table I shows the codewords for each symbol, the indexes of k length codeword and r th word of length are useful for hardware design that will be addressed in the next section.

III. SYSTEM ARCHITECTURE OF THE PROPOSED VLC

Figure 4 shows the system diagram of the proposed architecture. The input symbols are with fixed length, however, the output codeword is variable length, so a first-in first-out (FIFO) memory is required as a speed buffer. The symbols are written to FIFO by fixed clock rate. When the overflow-flag (OF) of FIFO becomes high, the write-data of FIFO should be idle. In another

FIGURE 3 The fast codeword creation.

FIGURE 4 The system diagram of proposed real-time VLC.

case, as the FIFO is empty, the underflow-flag (UF) of FIFO becomes high, this case denotes that the bit-stream of output is not available now.

The symbols could be encoded to the k th length codeword and the r th word of k th length using Length Encoder and Order Encoder respectively, which can be realized by referring Table I. The codeword length k is loaded to the down-counter and the Length Decoder (LD) in the same time. The LD controls which-one modules of k th-length codeword to be selected by enable pin (En). The Bi is binary code that denotes the r th word of length, where $i = 0$ to m , m is the most significant bit. With

the value of Bi and En , the codeword can be selected to load the shift register ($R1 \sim Rk$). Now, we can attain one-bit codeword from the shift register in every one clock. As coding procedure goes on, the value of down-counter is decreased by one when shift register outputs one-bit codeword. Until the value of down-counter becomes zero, the read-clock of FIFO is active, then the new symbol will be processed by the same procedures.

Figure 5 shows the internal circuits of k -length modules, which is easy to realize by using the Table I. The modules of codeword banks are generated by tri-state buffers, in which the k th bit codeword only use k buffers. The detailed scheduling

FIGURE 5 Codeword bank in the modular of k -bits.

of this architecture for real-time coding symbols of h , a , e , b is shown in Table II. The two-phase $F_{\phi 1}$ and $F_{\phi 2}$ are used to control the down-counter and the shift register, respectively. The timing diagram is shown in Figure 6. When coding procedure is starting, the ' h ' symbol first inputs the coding system from FIFO, then the Length Encoder and Order Encoder individually outputs $k=5$ and $B_{1 \sim 0}=10$, because its word-length and order is 5 and 3 respectively. The k value sends to Length Decoder (LD), then we attain the $E_{4 \sim 1}=1000$ from LD. In next phase, the 5th bank codeword is loaded to shift register. The shift register data is

variable from $R1 \sim R5$, now the data is '10110'. Thus we first get '1' in the output of shift register in the first clock. In the second clock, the shift register outputs '0', and the down counter becomes to 4. Continue this procedure, we should completely attain the bit stream '10110' for h symbol after 5 clocks. Then the down counter becomes to zero, we again read FIFO data for VLC coding. The new symbol ' a ' enters this architecture, which is coded with the same above procedures. Hence we spend k clocks to code k th length codeword, there is no waiting cycle, which can achieve a real-time operation.

TABLE II The operation scheduling of proposed VLC architecture

Clock time	Input symbol	FIFO read	Down- counter	Encoder		Decoder $E_{4 \sim 1}$	Shift reg load	Output bit
				k	$B_1 \sim 0$			
1	h	$F_{\phi 1} \uparrow$	5	5	10	1000	$F_{\phi 2} \uparrow$	1
2	h	$\times$	4	5	10	1000	$\times$	0
3	h	$\times$	3	5	10	1000	$\times$	1
4	h	$\times$	2	5	10	1000	$\times$	1
5	h	$\times$	1	5	10	1000	$\times$	0
6	a	$F_{\phi 1} \uparrow$	$0 \rightarrow 2$	2	00	0001	$F_{\phi 2} \uparrow$	0
7	a	$\times$	1	2	00	0001	$\times$	0
8	e	$F_{\phi 1} \uparrow$	$0 \rightarrow 4$	4	01	0100	$F_{\phi 2} \uparrow$	1
9	e	$\times$	3	4	01	0100	$\times$	0
10	e	$\times$	2	4	01	0100	$\times$	0
11	e	$\times$	1	4	01	0100	$\times$	1
12	b	$F_{\phi 1} \uparrow$	$0 \rightarrow 3$	3	00	0010	$F_{\phi 2} \uparrow$	0
13	b	$\times$	2	3	00	0010	$\times$	1
14	b	$\times$	1	3	00	0010	$\times$	0

FIGURE 6 The timing diagram of proposed VLC architecture.

In this architecture, the VLC coding scheme is partitioned into two stages, one is the codeword length and order index for bit control, another is the codeword bank for actual codeword generation. In the codeword banks, we only employ a few

tri-state buffers rather than memory devices such as ROM, PLA and so on. Actually, we can use one pass transistor instead of one tri-state buffer in physical design. Furthermore, some bit could be directly tied to VCC or GND in codeword banks,

so the circuits should be further reduced in our architecture. In this system, the control circuits involve the Length Encoder, Order Encoder, Length Decoder and down counter. Length Encoder encodes the number of bit for each codeword, which combines with down counter to control the bit stream data. So Length Decoder and down counter are also required even for ROM based or PLA based VLC coding. Order Encoder and Length Decoder are overhead in our architecture, the circuit complexity will be discussed in the next section.

VI. PERFORMANCE COMPARISON

The conventional VLC coding always uses the ROMs [4] or PLAs [6] to build-in codeword table. Generally, the circuits of ROM or PLA contain the AND plane and OR plane. In ROM device, the circuit size of AND plane is fixed, the cell location is called its address. Assumed that maximum resolution is n bits for symbols, the number of address is 2^n and each address contains n bits, and the circuits need use two transistors to represent '0' or '1'. Thus the number of transistors requires $n2^{n+1}$ in AND plane. In OR plane, we require $p2^n$, where p is maximum codeword length. So, the sum of the number of transistor S_{ROM} needs

$$S_{\text{ROM}} = n2^{n+1} + p2^n = 2^n(2n + p). \quad (3)$$

To reduce the circuits of ROM based, the PLA device is always employed for VLC coding. Since the AND plane and OR plane of PLAs are all

programmable, the total number of transistors requires $2mn$ and mp for AND and OR plane respectively, where m denotes the total number of input symbols. So the number of total transistor S_{PLA} in PLA based requires

$$S_{\text{PLA}} = 2mn + mp = m(2n + p). \quad (4)$$

Since $2^n > m$, the circuit of PLA based is less than the one of ROM based.

In the proposed architecture, the codeword banks contain k groups, the k th group uses k transistors if all bits need pass transistor. So the maximum number requires $\sum_{i=2}^p i$ transistors in codeword banks. In Length Decoder (LD), we use $E1 \sim Ek$ to control which one codeword banks is active, in which we can find $E_{\text{max}} = p$. Thus, the number of output requires p and the number of maximum input is $\log_2^p$. We need $2p \log_2^p$ transistors to realize the LD circuit. Consider the Order Encoder, the number of output Bi is less or equal to $\log_2^p$, then we need $2m \log_2^p$ transistors in the maximum condition. Therefore, the number of transistor requires

$$S_{\text{proposed}} = \sum_{i=2}^p i + 2p \log_2^p + 2m \log_2^p \quad (5)$$

in proposed architecture. Table III shows the comparison of the transistor number with ROM based, PLA based and the proposed architecture. The complexity of our method is about 1/4 and 1/2 of ROMs and PLAs in average, respectively. Survey the complexity order, we can find the one

TABLE III The performance comparison with ROM, PLA and proposed method

Circuits	ROM [4]			PAL [6]			Proposed		
	AND plane	OR plane	Total number	AND plane	OR plane	Total number	Word banks	LD and OD	Total number
General [#]	$n2^{n+1}$	$p2^n$	$2^n(2n+p)$	$2mn$	mp	$m(2n+p)$	$\sum_{i=2}^p i$	$2(m+p) \log_2^p$	$\sum_{i=2}^p i + 2(m+p) \log_2^p$
$n=8, m=160, p=24$	4096	6144	10240	2560	3840	6400	309	1687	1996
$n=7, m=80, p=16$	1792	2048	3840	1120	1280	2400	145	768	913
$n=6, m=40, p=12$	768	768	1536	480	480	960	77	372	449
$n=5, m=20, p=10$	320	320	640	200	200	400	54	199	253
Average	1744	2320	4064	1090	1450	2540	146	757	903

[#] The number of symbols is m , each symbol has n bits. The maximum length of codeword is p .

of ROM based is the exponential-product order $O(2^n P)$, the one of PLD based is the linear-product order $O(mp)$, and the one of proposed architecture is the log-product order $O(m \log_2^p)$. It is obvious that the proposed architecture becomes more efficient when the codeword size is large.

V. CONCLUSIONS

In this paper, we first present an efficient VLC method to create the codeword quickly, and the constructed codeword is capable to compress data with variable length format. With these codewords, the architecture of proposed VLC is partitioned to two parts, one is an index for coding control, and another is a real codeword bank. With this scheme, the complexity order of circuits can be efficiently decreased from the linear-product order to log-product order, so the proposed architecture becomes popular. When the codeword size is more and more large, the proposed technique becomes more efficient than the conventional ROM or PLD based. The timing shows that we use k clocks to encode in k th-length codeword, there is no waiting cycle, so this architecture can achieve a real-time purpose. As this proposed VLC architecture combines with other coding methods such as MPEG or JPEG coding, we can take full system simulation. It becomes higher efficiency when system on one-chip (SOC) is developed in the future.

References

- [1] Bassiouni, M., "Data compression in scientific and statistical data based", *IEEE Trans. Software Eng.*, **SE-11**, 1047–1058, Oct., 1985.
- [2] Gallager, R. (1978). "Variations on a theme by Huffman", *IEEE Trans. Inform. Theory*, **IT-24**, 668–674.
- [3] Ziv, J. and Lempel, A. (1977). "A universal algorithms for sequential data compression", *IEEE Trans. Inform. Theory*, **IT-23**, 337–343.
- [4] Shih-Fu Chuang and David G. Messerschmitt, "Designing high-throughput VLC decoder Part I- Concurrent VLSI architectures", *IEEE Trans. Circuits Syst. Video Technol.*, **2**(2), 187–196, June, 1992.
- [5] Amar Mukherjee, Ranganathan, N. and Bassiouni, M., "Efficient VLSI Designs for Data Transformation of Tree-Based Codes", *IEEE Transactions on Circuit and Systems*, **38**(3), 306–313, March, 1991.
- [6] Sun, M.-T. and Lei, S.-M. (1990). "A parallel variable length code decoder for advanced television applications", *Signal Processing of HDTV*, II, L. Chiariglione, Elsevier Science Publishers.
- [7] Park, H. and Prasanna, V. K., "Area efficient VLSI architectures for Huffman coding", *IEEE Transactions on Circuit and Systems-II*, **40**(9), 568–575, Sep., 1991.
- [8] Hsieh, C. T., "The systematic approach for concurrent VLC decoder", *IEEE Transactions on Consumer Electronics*, **43**(3), 918–924, Aug., 1991.
- [9] Lee, Y. S., Jong, J. J., Hsu, L. C., Jaw, M. Y. and Lee, C. Y., "A memory-based architecture for very high throughput variable length codec design", *1997 IEEE International Symposium on Circuit and Systems*, pp. 2096–2099.
- [10] Fukuzawa, Y., Hasegawa, K., Hanaki, H. and Iwata, E., "A programming VLC architecture for video compression DSP", *1997 Signal Processing, SIPS-97*, pp. 469–478.

BIOGRAPHIES

Shih-Chang Hsia received the Ph.D. degrees from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, ROC, in 1997. During 1986–1989, he was an engineer in the R&D department of Microtek International Inc., Hsin-Chu. He was an instructor and associate professor in the Department of Electronic Engineering, Chung Chou Institute of Technology during 1991–1997. Since 1997, he is an associate professor in Department of Computer and Communication Engineering, National Kaohsiung First University of Science and Technology. His research interests include VLSI design for HDTV systems, video coding and processing, data hiding system.

Chien-Cheng Tseng received the B.S. degree, with honors, from Tatung Institute of Technology, Taipei, in 1988, and M.S. and Ph.D. degrees from National Taiwan University, Taipei, in 1990 and 1994, respectively, all in electrical engineering. From 1995 to 1997, he was an associate researcher at Telecommunication Laboratories, Chunghwa Telecom Co., Ltd. in Taoyuan, Taiwan. He is currently an assistant professor in the Department of Computer and Communication Engineering at National Kaohsiung First University of Science and Technology. Dr. Tseng is a member of IEEE. His research interests include digital signal processing, pattern recognition, and electronic commerce.