A Memory-Efficient VLC Decoder Architecture for MPEG-2 Application

Seung-joon Lee, Ki-bum Suh, Jong-wha Chong

CAD & Communication Circuit Lab., Dept. of Electronic Engineering
Hanyang University, Seoul, Korea
Phone: +82-2-2290-0558, FAX: +82-2-2299-2129
e-mail address: sjlee@shira.hanyang.ac.kr

Abstract

Video data compression is a major key technology in the field of multimedia applications. Variable-length coding is the most popular data compression technique which has been used in many data compression standards, such as JPEG, MPEG and image data compression standards, etc. In this paper, we present memory efficient VLC decoder architecture for MPEG-2 application which can achieve small memory space and higher throughput. To reduce the memory size, we propose a new grouping, remainder generation method and merged lookup table (LUT) for variable length decoders (VLD's). In the MPEG-2, the discrete cosine transform (DCT) coefficient table zero and one are mapped onto one memory whose space requirement has been minimized by using efficient memory mapping strategy. The proposed memory size is only 256 words in spite of mapping two DCT coefficient tables.

I. Introduction

Variable length coding or known as Huffman coding [1] is a widely used technique in video compression systems. The main idea of variable length coding is to remove statistical redundancy after DCT and quantization. Variable length code (VLC) encoding is table look-up process with the mapping function between source symbols and variable length codewords which is according to the statistics of input symbols. Shorter codewords are assigned to frequently occurring symbols, and vice versa, so that minimum average codeword length and bit rate reduction can be achieved.

For maximum compression, the coded data is normally sent through a continuous stream of bits with no specific guard bit assigned to separate between two consecutive symbols. As a result, decoding procedure in this case must recognize the code length as well as the symbol itself.

In the MPEG-2 standard, the variable length code stream is composed of many types of macroblock-level data, resulting in 16 different VLC tables and a total of about 450 codewords. The DCT run-level coefficients (DCT coefficient table zero and DCT coefficient table one) are most frequent, accounting for more than 80% of the variable length bit stream.

In the MPEG-2, the size of VLC codebook for DCT coefficients table zero is 113 words and the size of VLC codebook for DCT coefficients table one is 114 words, but whose bit range is from 1 to 16bits without sign bit. The VLC decoder is mandatory to match all codewords in the codebook, unless a properly structured technique is adopted to allocate the symbols in the memory. The look-up memory size could reach 65536($=2^{16}$) words by direct mapping all possible codewords. It means waste in the memory resource and lower efficiency in symbol search.

This paper describes three approaches which can accomplish faster VLC decoder with efficient memory. To reduce memory resource, We propose effective LUT, group generation method and new remainder approach for MPEG-2 intra table (DCT coefficient table one) and MPEG-1 table (DCT coefficient table zero).

In Section II, some previous works are introduced, and in Section III, we propose the three approach methods for effective memory in VLC decoder. The system architectures based on new approaches are illustrated in Section IV, and Section V is the conclusion section.

II. Background

Various approaches have been presented to implement high throughput VLD's. They can be largely divided into two categories. Two categories are a binary tree search algorithm and a bit-parallel method. Traditionally, VLC decoding is implemented through tree searching algorithm. The binary tree search algorithm can be implemented based on a Huffman tree, which uses the principle of a token propagation in a reverse binary tree constructed from the original codes. However this approach is not suitable for high performance systems since it can decode only 1 bit per cycle. The newer approach is bit-parallel decoder which processes a codeword per one clock regardless of its length.

In case of bit-parallel decoder, the memory efficiency is a considerably important factor. In cases where the number of codewords in the table is large, there are some bits that are common to the long VLC's, which we call prefix. By exploiting these common prefixes, the size of the LUT can be reduced. Several approaches have exploited this prefix predecoding method to efficiently decode the VLC's [2], [3], [4]. In other words, the VLC decoding is simplified into two step processing. First, the common bit pattern in a codeword is recognized or a fixed amount of bits is processed in the bit stream, then they are used as the first reference for the memory searching. After the same bit pattern or fixed amount of bits are removed, the remaining bits in the codeword are used for the second reference for the memory searching. Hsieh and Kim [2] partition the codewords into maximum likely bit pattern (MLBP) and remainder. The VLC codebook for only MPEG-2 intra block (DCT coefficient table one) are grouped by MLBP. For the codewords with same length, it partitions them into groups based on the MLBP. Basically, it tries

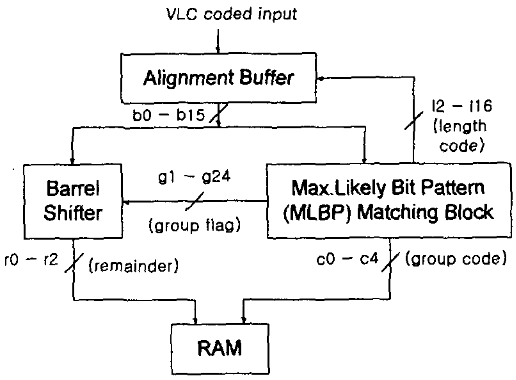

Fig. 1: System block diagram for DCT coefficient table one proposed by Hsieh and Kim [2]

to extract the same bit pattern within the codewords of the same length as the first reference (group numbers) for the memory searching. After extracting MLBP from the codeword, the remaining bits are called remainder which is second reference to access symbol memory. In the previous paper proposed by Hsieh and Kim, total group's number used as the first reference is 24 ($\approx 2^5$) and the lengths of the largest remainders are 3 bits ($=2^3$), so that required memory size is 256 words for one DCT coefficient table. As a result, in the previous architecture, 512 words memory space is needed to map two DCT coefficient tables. Fig. 1 is a System block diagram proposed by Hsieh and Kim.

III. Merged LUT, New Group Numbering and Remainder Generation Algorithm

The merged LUT, new group numbering and remainder generation method are presented in this section for improving the efficiency of memory utilization in VLC decoding architecture.

If we map two DCT coefficient tables onto one memory space, some factors are needed to be considered. First, in order to use small memory space, not two LUTs but a merged LUT for two DCT coefficient tables must be generated as shown in Table 1. And then, two signals should be considered. First signal, MPEG2_intra, is used to differentiate DCT coefficient table zero from table one. Second signal, First_coefficient, is used to classify first (DC) coefficient in DCT coefficient

Length	VLC Table	First_ coeff.	MPEG2_intra	Group no.	MLBP	Rem	Run	level	priority
1	1	1	0	1	1	X	0	1	A
2	10	0	0	2	1	000	End of Block		a
	10	X	1			001	0	1	
	11	0	0			100	1	1	
3	011	X	0	3	011	X	1	1	B
	010	X	1	4	X10	0	1	1	
	110	X	1			1	0	2	
4	0100	X	0	5	01	000	0	2	b
	0101	X	0			010	2	1	
	0110	X	1			101	End of Block		
	0111	X	1			111	0	3	
5	00101	X	0	6	001	010	0	3	c1, c2
	00110	X	0			100	4	1	
	00101	X	1			011	2	1	
	00110	X	1			101	1	2	
	00111	X	X			110	3	1	
	11100	X	1	7	1110	0	0	4	A
	11101	X	1			1	0	5	
⋮									
16	0000000000011000	X	X	32	0000000000011	000	13	2	
	0000000000011001	X	X			001	12	2	
	0000000000011010	X	X			010	11	2	
	0000000000011011	X	X			011	31	1	
	0000000000011100	X	X			100	30	1	
	0000000000011101	X	X			101	29	1	
	0000000000011110	X	X			110	28	1	
	0000000000011111	X	X			111	27	1	

Table 1: A MPEG-2 VLC table for DCT coefficient table zero and table one, where x means don't care

table zero. The result of the analysis of the two tables shows that 64% of them are exactly the same. This same parts are marked as don't care(X) in 4th column (MPEG2_intra) of Table 1. From the result of these analysis, we know that a merged LUT size can be reduced. Basically, the generation of MLBP follows previous method proposed by Hsieh and Kim. After generating a merged LUT for table zero and table one, it partitions codewords into the same length (1st column in Table 1), and then, for the codewords with same length, it partitions them into MLBP (Maximum Likely Bit Pattern) with same bit pattern. Group numbers used as the first reference to access memory are generated by 3 factors which are MLBP (6th column in Table 1), First_coefficient (3rd column in Table 1), MPEG2_intra (4th column in Table 1). As above mentioned, in the case of a merged LUT, MPEG2_intra bit are necessarily needed to differentiate DCT coefficient table zero from table one. After extraction MLBP from the codeword, remaining bits are called remainder which is the second reference for the memory searching. In general case, same group number is given, when two factors (MLBP, MPEG2_intra) are correctly matched (e.g. group 3, group 4, group 7, group 11 ~ 22, group 24 ~ 26), in another case, only one factor (MLBP) is needed for group numbering (e.g. group 2, group 5, group 6, group 8 ~ 10, group 23, group 27 ~ 32), and in third case, three factor (MLBP, First_coefficient, MPEG2_intra) are used for group numbering (e.g. group 1) as shown in Fig. 2. In this grouping strategy, some group have special case (group 2, 5, 6, 9, 10). An example as shown in Table 1, though group 5 must be divided by two groups (when MPEG2_intra is 0 or 1.) in fact, only one group number is given for group 5. Instead of allowing one group number, new remainder generation method is used and it generates new remainder bits which are shadowed area in Table 1. One or two bits for MPEG2_intra and First_coefficient are attached in LSB of remainder. This means that this algorithm uses unused address in the previous architecture. And also, this improves the efficiency of memory. The lengths of the remainders vary from 0 to 3 bits which can specified by the designer in terms of the tradeoff between codeword matching circuits and memory. Group number and remainder are combined to be used as the address for the symbol memory. In grouping, some group's MLBPs are the subsets of other groups'. The group 5's MLBP (01) is a

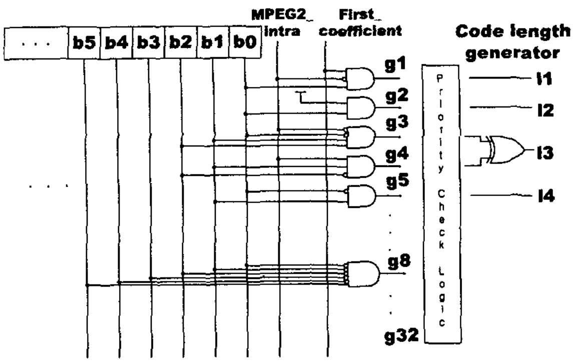

Fig. 2 MLBP matching block

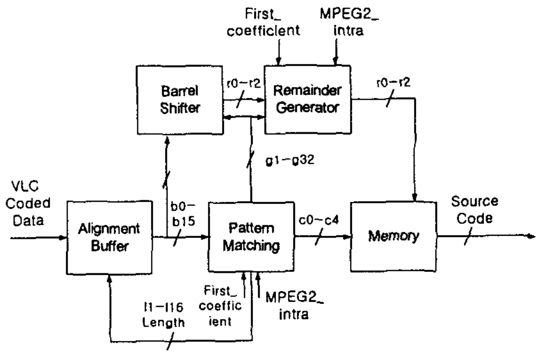

Fig. 3 New VLC decoder system architecture

subset of group 3's (011) that means whenever a group 3 codeword is recognized, the matched flag from both groups will be high simultaneously. This situation is not desired so that a priority check is needed to mask the lower priority group flag. One with higher priority is represented by a upper case character as shown in Table 1 (last column of Table 1).

IV. New VLC Decoder System Architecture

The new VLC decoder architecture is shown in Fig. 3. The input is adjusted by the alignment buffer such that the first bit of the VLC code is aligned to b_0 which is the 1st input bit to barrel shifter and MLBP matching block. The MLBP matching block, as shown in Fig. 2, generates three variables: first variable, group flags (g_1 - g_{32}) to control the barrel shifter for extracting the remainder (r_0 - r_3), second variable, length flags (l_2 -l_{16}) to feed back the alignment buffer for aligning the next VLC codeword, and third variable, group code (c_0 - c_4) for decoding RAM. The barrel shifter outputs 3 bit remainder from a 16 bit input which is controlled by the group flags (g_1 - g_{32}). The memory is a 256 word RAM which can be accessed by the combination of 3 bit remainder (r_0 - r_3) and 5 bit group code (c_0 - c_4).

V. Conclusion

A new memory efficient VLC decoder for MPEG2 application is presented. The proposed algorithm is based on a merged LUT, new group numbering and remainder generation algorithm. In the presented architecture, only 256 words memory space and one MLBP matching block are needed, whereas 512 words size memory block and two MLBP matching blocks have been used to decode two DCT coefficient tables in the previous architecture. There is no performance degradation compared with the previous architecture [2].

References

[1] D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proc. IRE*, vol. 40, pp. 1098-1101, Sept. 1952.

[2] Cheng-Teh Hsieh and S. P. Kim, "A concurrent memory-efficient VLC decoder for MPEG applications," *IEEE Trans. on Consumer Electronics*, 42(3):439-446, Aug. 1996.

[3] S. B. Choi and M. H. Lee, "High speed pattern matching for a fast Huffman decoder," *IEEE Trans. Consumer Electronics*, vol. 41, pp. 97~103, Feb. 1995.

[4] S. F. Chang and D. G. Messerschmitt, "Designing high-throughput VLC decoder Part I-Concurrent VLSI architectures," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 187-196, June 1992.